CREATING A VIRTUAL ENVIRONMENT FOR INTERACTIVE LEARNING

A Thesis Submitted
in Partial Fulfillment of the Requirements
for the Degree of
Master of Technology

by

K. V. Vihari

to the

Department of Computer Science & Engineering
Indian Institute of Technology Kanpur
May, 1998

Certificate

Certified that the work contained in the thesis entitled "Creating a Virtual Environment for Interactive Learning", by Mr.K. V. Vihari, has been carried out under my supervision and that this work has not been submitted elsewhere for a degree.

Dheeraj Sanghi

(Dr. Dheeraj Sanghi)
Assistant Professor,
Dept. of Computer Science and Engineering,
IIT Kanpur.

May, 1998

Acknowledgments

Giving me freedom to explore new fields, freedom to work at my own pace independently, extending considerate administrative support, urging me to go that extra inch on the way to perfection, directing me towards new opportunities, being flexible and reasonable – that was Dr. Dheeraj Sanghi, my guide. Gratitude to him flows naturally for allowing me to work in the area of my interest, for encouraging me to publish two papers, for helping me finish the thesis work satisfactorily, for seeking a sponsorship for my project.

My gratitude goes to Dr. T.V.Prabhakar for kindly agreeing to be my administrative supervisor in the absence of my guide. More than that, a heart-felt "thank you" to him for being so affectionate and encouraging.

I am also indebted to Dr. Vijay Gupta, Dr. Sanjeev Kumar, Dr. Deepak Gupta, my thesis examiners, for their valuable comments about the work.

Thanks also to Mahindra-British Telecom Limited, Mumbai, for sponsoring my project and for the lovely 2-day trip to Mumbai for project discussions. I am grateful to Dr. S.V.Albal for his inputs that have helped to refine the focus of this work and also for being such a considerate host and warm human being.

Special thanks to each of my classmates for their support in all the extra-curricular and academic activities I undertook.

For making my stay at Kanpur one of the most enjoyable and educative periods of my life yet, I will have to thank so many people as to make this page too small for that. Instead I will thank The Eternal Goodness, the One Source behind all, for Its unconditional benevolence, unfailing support through friends and Nature, and invaluable guidance through experiences.

Abstract

The Internet is being increasingly used for the process of information delivery in distance education. There has been a considerable amount of research in *asynchronous* methods of information delivery over the Internet, while comparatively little work has been done with *synchronous* information delivery mechanisms. This work focuses on synchronous information delivery mechanisms and studies some of the problems that can hinder effective Internet-based education. It puts forward the concept of "Virtual Environments for Interactive Learning" that aims to solve these problems. We propose a methodology based on this concept to develop distance learning software. The design and implementation of the concept in a software prototype called Virtual Lecture Hall are detailed. A preliminary evaluation of the work is also made.

Contents

Chapter 1

Introduction

With the rapid growth of Internet technology and Internet use, research in the field of distance education using the Internet has received a fillip. New and innovative uses of the Internet for information delivery in various learning environments are being explored. Virtual Classrooms, Virtual Universities and Virtual Workshops are springing up at various locations on the Internet.

Information delivery over the Internet can take two forms: synchronous and asynchronous. The difference between these two forms is based on the time gap involved between the processes of information sending by the instructor and information reception by the learner. In *asynchronous* delivery of information, the instructor makes information available at a point of time which the learner uses at a later point of time. An example of asynchronous delivery of information: preparing a hypertext/hypermedia website that a student can browse at his own pace. In *synchronous* delivery, the instructor and the learner participate in the process of education at the same point of time. An example of synchronous delivery: a "lecture" by an instructor sitting at his/her computer to a group of distant students connected through the Internet.

A survey of the state-of-the-art revealed a relatively large amount of work that addresses asynchronous information delivery, while the area of synchronous information delivery has not been explored significantly.

This work addresses the problem of synchronous delivery of information over the Internet, studying the issues that arise therefrom and attempting a solution to those issues. The motivation for attempting to study and solve this problem is the opportunity to work

with a fairly new and rarely-used paradigm of distance learning. The issues that arise out of synchronous delivery of information using multimedia tools across the Internet, could prove interesting to study and challenging to solve. A successful solution resulting out of this work could be a useful step forward for the fields of distance and continuing education over the Internet.

1.1 Our Contribution

It was found that, given a set of multimedia tools performing reasonably well over the Internet, a number of new issues arise during the process of information delivery. These issues have more to do with human factors in computing rather than the underlying technology. They reduce the overall effectiveness of the information delivery process.

The major thrust of this work is to address these issues as a means of improving the effectiveness of distance education using the Internet. The end-product of the work is a prototype software consisting of two sets of interfaces to be used at the instructor and learner sites. These interfaces are connected to each other over the network and function in tandem to take care of some of the human factors that hinder Internet-based education.

The work described in interdisciplinary in nature. We have borrowed ideas and concepts from the fields of Interfaces, Multimedia, Computer-Human Interaction, Learning Theory and Media Spaces.

1.2 Organization of the Report

The rest of this report is organized as follows:

- Chapter 2 presents a summary of the related work in the field of Internet-based education delivery. It also mentions some findings from related fields that have aided the development of our solution to the problems of synchronous information delivery over the Internet.
- Chapter 3 presents and explains the *VEIL* concept that forms the basis of our solution. It also presents a methodology that we propose for building effective Internet-based learning software.

- Chapter 4 gives of the design of the (*Virtual Lecture Hall*) software prototype that was built based on the *VEIL* concept and methodology.
- Chapter 5 outlines the implementation details of the *Virtual Lecture Hall* prototype, including some details about a simple network protocol designed for the exchange and processing of messages between the client- and server- sides of the *VLH* software.
- Chapter 6 summarizes the work and presents a preliminary evaluation of the prototype built. It also identifies some areas of future work.

Chapter 2

Related Work

In this chapter we present related work by researchers in the area of Internet-based learning. We present the details of the related work under two sections: (i) Internet-based learning: State-of-the-art, and (ii) Issues that affect effective information delivery. We also summarize some observations from the related fields of computer-supported co-operative work, media spaces, learning theory and interface, that have influenced the solution we proposed. A detailed report of the literature survey activity, containing many useful links to sites on the World-Wide Web, is included as Appendix A of this report.

2.1 Internet-based Learning: State-of-the-art

A substantial number of papers are available in computer-based education literature describing the experiences of researchers who used the Internet to create various types of virtual teaching/learning environments and/or to assist normal classroom environments [4], [3], [6], [24], [7], [9], [8], [25], [2], [12]. The works reported seek to simulate some of the basic functionalities of traditional learning environments (such as a classroom, a workshop, a laboratory, a tutorial, an instructor's office) via the Internet.

Some of the papers describe the use of Internet technologies to enhance normal classroom functioning and to improve the quality of classroom activities such as assignment giving/submission/correction [19], [9], [21], [13], [28]. [16] discusses the possibility of, and the interactions that can result in, the creation of learning environments that are different from normal classrooms ('virtual learning communities'). [26] reports of work done in providing 'situated learning' – learning in which a social environment is integrated into the

software to promote student (emotional) involvement. The 'situation' for the learning provided in [26] is not reported to be modeled *intentionally* [22] upon a particular real-world environment, but it comes close to a tutorial class.

The information delivery activities involved in most of these are asynchronous. Interaction between the instructor and student is confined to e-mail or bulletin boards. Two references [7], [26] report work that explores the use of the audio medium for the actual delivery of the teaching material, but again in the asynchronous mode. It is observed that, as such, not much work has been done to use the audio medium for synchronous information delivery. Where audio was used at all, it was done in conjunction with video. But we have not considered the use of video since it is not viable in the present Indian scenario.

As far as the state-of-the-art of the tools and technologies that support Internet-based information delivery was concerned, a search over the World-Wide Web revealed the presence of a large number of multimedia and other useful tools such as those for audio communication [29], shared text chat, shared whiteboard drawing, slide presentation. Many of these are present in shareware/freeware domain. Notable among these are: NetMeeting [20], SpeakFreely [30], PointPlus [27].

2.2 Issues that Affect Effective Information Delivery

Some work has been done to study the issues involved in using the Internet technology in the teaching process [11] [14], [25]. Some of these findings are to do with the precautions to be taken by instructors in preparing and presenting their material unambiguously in the new learning environments. An example of this is the problem of using the right metaphors in organizing the course websites. They also discuss the technological problems and "human factors" that the distant learner faces sitting in front of a computer and communicating through it with humans located somewhere else. Some of the problems mentioned are the feelings of isolation, loss of orientation, and loss of motivation, resulting from the impersonal nature of the interfaces.

These researchers have limited themselves to asynchronous learning for the most part when talking about information delivery mechanisms. They do not talk about the issues that arise out of using Multimedia tools especially the tools for audio or the video media. They rarely refer to communication in synchronous mode. Furthermore, they do not make observations about the dynamics of a group of people communicating in a new environment.

These issues have been studied by work in the fields of Computer-Supported Co-operative Work (CSCW) and Media Spaces [1].

In these fields, Bly et al and Ackerman have done very interesting and useful work [5], [1]. They provide insights into the suitability, advantages and drawbacks of using audio-only, and audio in combination with video, for everyday workplace communication. They conducted field-studies ranging over substantial periods of time. The work reported by [1] does not make use of computers at all, while the use of computers in [5] is not for communication procedures as such. So the interfaces that they use are of a different nature from the interfaces that a computer-based education application would require. But they do make many useful observations about the norms of interaction and the basic nature of the interfaces from the point of view of a user of the media. These observations are briefly mentioned in the next section. They give some guidelines that could influence the design of interfaces systems that support real-time communication. Some of these findings are listed as design guidelines in Chapter 4.

Further guidance about the design of interfaces comes from [10]. In this, a reflective, user-oriented design process called UCD (User Centered Design) is advocated. Some of the guidelines for UCD are also listed in Chapter 4. Going one step further with user-centered design is learner-centered design (LCD) of software. LCD aims to augment user-centered design by providing for 'scaffolding' (integrating tools that encourage the user to investigate and learn outside the regular classroom activities).

2.3 Some Observations from Related Fields

Some of the problems faced in Internet-based synchronous information delivery are very similar to the problems faced in CSCW (Computer Supported Co-operative Work) environments. Here, we briefly mention some of the observations about such environments.

Research in Media Spaces shows that when using electronic media for communication to support collaborate work, a particular *social environment* is created. For example, in the Thunderwire system reported by Ackerman [1], a group of colleagues working on video editing project, shared a broadcast intercom. Each person has a headphone set with a

[1] A Media Space is defined as an electronic setting in which groups of people can work together, even when they are not resident in the same place. In a media space, people can create real-time visual and acoustic environments that span physically separate areas. A media space causes communication that helps to create a social environment.

microphone attached to it. The primary purpose of this audio-only system is to enable them to talk to each other to exchange work-related information. However, it was found that over time and use of the system a social environment evolved. The users of the system came to depend on the system not just as a tool aiding them in their work, but as an emotional input that motivated them to use the system more frequently and to collaborate more.

The social environment, thus, plays an important part in determining the success of the communication as a whole. Where the social environment causes a positive emotional involvement in the persons using it, as it happened in the Thunderwire system, they tend to use the system more, contributing to the overall success of the system.

The *norms* [2] of such an environment and how the norms are followed by the persons taking part in the communication, play a pivotal role in determining whether it causes a positive emotional involvement or not. For example, there were moments of embarassment for two users of the Thunderwire system when they answered personal phone calls while others could hear them. There was no norm that others were supposed to follow in such a situation. This led the two users to reduce their use of the system. Following the norm about not answering telephones with the headset and microphone on, would have prevented this.

Further, the research also points out that the **interfaces** provided as a part of the media space determine how the norms of the environment are followed by the persons using it. In the Thunderwire system, the only interface between the system and user that informed about the on/off status of the microphone was a small green light. This was easily missed by the users and caused them to ignore the norm about answering telephones. A better interface to show the status of the phone would have prevented the violation of the norm.

Research in the field of interfaces offers further guidelines about the design of interfaces. The design of interfaces should aim to project a metaphor. The desktop metaphor is a common metaphor used to group interfaces provided by the shell of the Windows95/NT operating system. [22] points out the primacy of projecting a metaphor homogeneously (that is, not mixing two metaphors) and completely (implementing all parts of the metaphor). This aids in making the interfaces more effective. The importance of implementing the interfaces so as to give an intuitive idea about the system's responses to the user is also pointed out. Further, the importance of user-centered design of interfaces is also advocated.

[2] A norm is defined as a group accepted mode of behavior; in other words, a behavior protocol that one follows when in an environment.

In this approach, the interface design is viewed as a process of successive refinement guided by system use. This helps to create interfaces most suitable for use in the environment.

These findings, together, form the basis for the concept of *Virtual Environment for Interactive Learning (VEIL)* that we propose as a solution for the issues that arise in using synchronous multimedia tools for Internet-based education. The *VEIL* concept is presented in the next chapter.

Chapter 3

The VEIL Concept and Methodology

In this chapter, the background for the development of the *VEIL* concept is briefly described. The *VEIL* concept that we propose to improve the effectiveness of synchronous information delivery process is presented and explained. We also propose a methodology that could be be used to implement the *VEIL* concept in distance learning software. The background for this methodology, its algorithm and the rationale behind the algorithm are also outlined.

3.1 Background

An Internet-based distance learning environment is similar to a CSCW environment in that in both places there is a use of electronic media for communication. It follows from the observations about media spaces mentioned in Section 2.3 that, for an Internet-based distance learning software package to be successful, it needs to provide more than just a set of tools for information delivery. The software needs to provide a social environment that creates a positive emotional involvement in the learning process for the learners. In other words, the environment should project the "presence" of the instructor and the other learners to each learner and give a "personal touch" so that the learner can concentrate on communicating with the instructor and the other learners rather than on the tools. This has a greater chance of motivating the learner. This provision for emotional involvement is called *situated learning* in the literature of Learning Theory. This emotional involvement and "personal touch" are natural to traditional learning environments like a classroom or

a lecture hall.

An emotional involvement is positive when the social environment makes the user feel comfortable with the norms of conduct of the environment. This ability to make the user comfortable with the norms is in turn dependent upon the interfaces that are provided along with the tools. Hence the interfaces provided along with the software hold the key for software the ensures effective information delivery.

A simple bundling of the tools and interfaces required for communication and information delivery over the Internet does not automatically ensure effectiveness. This could be due to one of two reasons. (1) The tools and their interfaces may not provide for a considerable social environment to be formed. (2) Even if a social environment emerges out of the bundling, the norms of conducting oneself may not be clear or may be counter-intuitive. Given this situation, the inclination of the user to use the software more frequently and motivatedly, for learning, reduces with time. The user may not use the software again, after the novelty of using the tools for the first time wears off.

3.2 VEIL Concept

Our solution for ensuring greater effectiveness is to make a *conscious* attempt to create a social environment. To ensure that this social environment provides a positive emotional involvement, we suggest that the environment be modeled upon a real-world environment (such as a classroom or a lecture hall). This involves not just providing the basic functionality of the real-world environment. Some of the researchers have done this, as is reflected in the names given to their software or projects: *Virtual Classroom, Virtual Workshop etc.*. Modeling the real-world environment also involves modeling the norms of the environment and designing interfaces that implement these norms. This designing of interfaces to implement norms forms the crux of the solution that we propose.

It is to be noted that there is an advantage in modeling the norms of a real-world environment rather than trying to define completely new ones. The norms of the real-world are usually developed as humans use them over time. They get modified and refined until a stable and comprehensive set of effective norms are hit upon. So using these real-world norms as a base set on which to model the norms of the virtual environment seems to be a justifiably good idea.

We call the environments that result from this comprehensive modeling of real-world

environments by the name of *Virtual Environments for Interactive Learning.* The environment is "virtual" because the people who participate in it are not face-to-face as in a traditional environment. Since it takes care of the human interactions that are an essential part of face-to-face learning, the virtual environment is "for Interactive Learning".

3.3 VEIL Methodology

We propose a methodology that could be used to create distance learning software that conforms to the *VEIL* idea in an organized and time-efficient manner. This methodology uses the finding about the availability of numerous multimedia tools off-the-shelf and advocates their use in an effort to avoid duplication of effort and reduce product cycle-time. For the most part, the interfaces that these tools provide are not sufficient to implement the norms of the learning environment. This methodology attempts to use the tools for the basic functional part of the information delivery process and advises the development of additional interfaces for the creation of learning environment. It suggests a way in which the interfaces can be homogeneously integrated with the tools to produce a *Virtual Environment for Interactive Learning.*

As a historical note it is to be stated that the methodology resulted out of an actual experience of designing and developing a *VEIL* software prototype called the "*Virtual Lecture Hall (VLH)*". The methodology is an abstraction of the steps taken and the practical problems addressed during the design and implementation phase of the *VLH* software.

We present the methodology here and use the details of *VLH* design and implementation to illustrate the various steps and decisions of the methodology in the next two chapters.

To begin with, we present a small background to highlight the problems in creating synchronous multimedia learning environments such as *VEILs*. A set of three guidelines that we evolved to help to solve these problems are presented next and the rationale behind the guidelines is explained subsequently. An algorithm that incorporates these guidelines, forming the core of the methodology is then presented as a simple seven-step algorithm [18].

3.3.1 Background

The focus on making distance learning software more effective led us to the use of synchronous or *live* [1] *multimedia.* It can be seen that the use of live multimedia is advisable,

and to some extent a necessity, when information has to be delivered primarily and effectively over a network to a group of remote learners.

However, the use of live multimedia brings with it a set of practical problems. Chief among them are:

1. the cost of hardware involved
2. the difficulty of setting up such a system at each learner-site,
3. the difficulty of having good connectivity and low-delay network throughput,
4. the cost and time involved in developing good-quality multimedia software, and
5. the problem of efficiently integrating the software for the various media for effective information delivery.

3.3.2 Guidelines

The solutions to these problems can be found as answers to the following questions:

- (a) For problems 1-3: what is the minimal set of multimedia tools that can be used for effective information-delivery?
- (b) For problem 4: what is the best way to build these multimedia tools?
- (c) For problem 5: what philosophy should guide the integration of the tools and what interfaces should be provided so that successful information delivery results?

3.3.3 Rationale

The rationale behind the above guidelines is as hereunder:

- A "minimal set of multimedia tools" is assumed to reduce the cost and setup burden of the infrastructure involved. For example, if a method and justification for not using one of the media can be found (which we tried to do for 'video' in the design of the *Virtual Lecture Hall*), a portion of the hardware costs and network bandwidth

[1] By "live" multimedia, we mean that a significant portion of the multimedia content is prepared and transmitted on-line.

burden could be lessened. Thus the answer to question (a) solves problems 1-3 to a certain extent. It is to be noted that problems 1 (hardware) and, especially, 3 (networking issues) are problems in their own right, and are not dealt with in this work. Minimizing the number of multimedia tools is also advocated because the presence of unnecessary tools and interfaces clutters the workspace and reduces the effectiveness of the metaphor that is projected by the system.

- The World-Wide Web (WWW) provides an answer to question (b). As pointed out in Section 2.1, a number of stand-alone tools for audio broadcast/communication, for shared white-boards, text chat etc. are available from the WWW. While some of these tools are marketed as costly products by some companies, many of them are available as freeware or shareware. In addition, some of the tools also provide the source code also to enable further modification. Some of these tools being the effort of groups of people for over substantial periods of time, are of good quality also. Use of these tools could cut down the cost and the time of developing distance learning software and avoids "re-inventing the wheel" [26].

- The answer to question 3 is essentially the creation of a *VEIL*. The creation of a VEIL provides a guiding philosophy to build additional interfaces and to integrate them so that effective information delivery results. The idea of modeling a real-world environment as mentioned under *VEIL Concept* subsection also aids in deciding the minimal set of tools to be used in the software.

3.3.4 Algorithm

The methodology we propose, that seeks to solve and answer the five problems and three questions mentionedn the previous subsections, is out-lined in the following algorithm.

1. **Requirements Analysis**: Analyze the educational/training needs which the software is going to address.

2. Choice of Real-World Model: Choose an appropriate real-world environment where the learning process as specified in Step 1 could be situated in.

3. **Determine** *Minimal Set of Media*: Analyze real-world environment in terms of its norms and the minimal sets of media that could be used to preserve the norms.

4. **Multimedia Software Collection**: Collect a large set of multimedia software tools that could be used over the Internet. The World-Wide Web is a very good source to acquire them.

5. **Decide Minimal Set of Tools**: Choose from the collection of multimedia software tools those that facilitate the use of the media identified as the minimal set of media in Step 3. While choosing the tools, some tools may need to be included to overcome the limitations of communicating in a virtual space. Also consider the hardware requirements and costs of using these tools while choosing them.

6. **Additional Interface Design and Implementation**: Design additional interfaces (the *VEIL* interfaces) to augment the interfaces provided by the available tools. These should help the users to follow the norms of the environment and to overcome the negative effects of physical isolation [11], [22] [10]. Some interfaces may also be required to help implement modifications to the real-world norms or to implement entirely new norms. Implement the interfaces, keeping in view the notion of "iterative refinement of interfaces through system use" proposed in [10].

7. **Integration**: Integrate the tools and interfaces with possible modifications to the tools, if required.

It is to be noted that the selection of an appropriate real-world model at Step 2 plays an important role in the algorithm. A correct choice would not only provide for effective learning, it also helps to minimize the number of tools that need to be used.

This methodology can be used to build distance learning software (choose tools, develop additional interfaces and then perform integration) at low software development costs. This building process can have typical development times of 3 to 6 months using some of the Rapid Application Development (RAD) tools like Microsoft Developer Studio.

Chapter 4

Virtual Lecture Hall Design

The design of an Internet-based software prototype called *Virtual Lecture Hall (VLH)* which implements the *VEIL* concept is discussed in this chapter. The guidelines used in designing, implementing the various tools and their interfaces are presented, followed by the details of the design.

4.1 Background

The steps followed in designing the *VLH* software prototype illustrate the algorithm of the *VEIL* methodology step-by-step. We, thus, present the background for the design of *VLH* under the headings of the steps of the algorithm.

4.1.1 Illustrating the VEIL Methodology

1. **Requirements Analysis**: The development of *VLH* had the following requirements specification. The targeted users of the software are a set of users who are familiar with using computers, who also have some technical background. One of these users delivers a lecture or conducts a seminar to share information with the others. The one who shares the information is referred to as the "instructor" and the others as "learners". Two real-world scenarios where this sort of environment would be created are: when a professor at a University conducts a course to continuing education students over the Internet, or when a corporate training team member conducts an in-house seminar series to update the skills of his colleagues working at physically

different locations. This later scenario was taken from an actual specification by Mahindra-British Telecom Limited, Mumbai, which sponsored the project of which this work is a part.

The instructor of such a group of people would prefer to conduct the seminar/lecture assisted by an Over-head Projector (OHP) for showing slides, rather than take a pedagogic approach like in a normal school classroom.

2. Choice of Real-World Model:

   The real-world model chosen to suit the basic requirements mentioned in Step 1 was that of a Lecture Hall. A Lecture Hall usually provides an audio communication medium and an OHP (and/or a blackboard). These are the basic requirements of an instructor like the continuing education professor or the corporate training team member. So a Virtual Lecture Hall environment that is modeled on the real-world Lecture Hall was to be built.

3. **Determine** *Minimal Set of Media*: Given the environment of a Lecture Hall, it was decided that the norms of the Lecture Hall could be preserved with the use of the audio, graphics and text media.

   Use of video medium could possibly help in the implementation of some of the norms. However, it was argued that the use of video was not really necessary. This is because in the environment of a seminar, the expressions of the instructor are not very essential to the basic functioning of the paradigm. (In fact, seminars that use a projector or OHP do take place under reduced lighting conditions also.) The voice and the material presented are sufficient in themselves in most cases in delivering an effective seminar/lecture.

4. **Multimedia Tool Collection**:

   This step of the methodology was performed as part of the literature survey work mentioned in Chapter 2. A number of software packages were downloaded from the WWW and tested.

5. **Decide Minimal Set of Tools**:

   Within the purview of the three media chosen in Step 3, a number of choices for software tools was present. Some of the tools that could be used were a shared whiteboard, a text chat, file transfer, slide presenter, shared application, collaborative

browsing and audio tools that provide broadcast or conference mode. Out of these, a choice had to be made so that the tools could help to implement the norms of the Virtual Lecture Hall environment to be created.

Even though it was decided that video was not fundamentally essential for the instructor to present the seminar, (s)he still needs some way to *see* that the audience wants to interact with him(her). (Perhaps there is a question to be asked by someone in the audience). The normal procedure in a real-world Lecture Hall for the members of the audience is to either directly (orally) interrupt the instructor or to raise a hand or make some visual signal to attract the instructor's attention. In the absence of a sense of direction in virtual space and the absence of video in the virtual environment, both these norms would be ineffective. The first one would leave the instructor wondering who interrupted him in cases where there is overlap [1] and the latter ones are simply useless in the absence of video. To overcome this problem, we decided to include a text-chat facility and also to modify some of the norms to facilitate interaction between the instructor and the audience in the Virtual Lecture Hall case. This, rather than adding video, was chosen because video is costly in terms of infrastructure, processing and usage complexity. Thus the *VLH* software prototype contains the following tools: audio tool (replacing the public address system of the Lecture Hall), whiteboard (instead of the blackboard), a slide presenter (instead of the OHP) and a text-chat (to help implement the norms).

For the audio tool, SpeakFreely by Fourmi Labs, was chosen for its audio broadcasting capability, its support for encryption, compression and other features. Microsoft NetMeeting was chosen for its packaging of the rest of the chosen tools in a convenient fashion. (Though NetMeeting supports audio conversation, it is only in the one-to-one mode. In the conference mode, audio is disabled.)

6. **Additional Interface Design and Implementation:**

After analyzing the norms of the Lecture Hall environment both in the real and virtual worlds, the details of simulating the norms were worked out. It was found that some norms needed slight modification and a few new norms were required. Interfaces to help implement these were designed. The conceptual details [17] of these are given in the sections to follow. The actual details of the interfaces that conform to the *VEIL* concept are mentioned in the next chapter on *VLH* implementation. These additional interfaces are referred to as *VEIL* interfaces. Some of the findings from the

survey work about Media Spaces and Interfaces were used as guidelines in designing the interfaces. There guidelines are listed in the immediately following section.

7. **Integration:**

   It was decided that the tools chosen for the basic functioning of the Virtual Lecture Hall would be used as they were. The tools could have been modified to link some of their functionality with the new interfaces designed and implemented. But since this task is time-consuming, it was not attempted.

   The task of integration, thus, involved packaging of the 3 components of the *VLH* prototype, namely: SpeakFreely, NetMeeting and the *VEIL Interfaces.*

4.2 Design Guidelines

This section lists some of the observations gathered from the work of researchers who worked with the audio medium and with interfaces. The following observations guided the design of the VLH prototype:

- In a real Lecture Hall, the extent to which the listener is captivated and remains so depends largely upon the instructor. This direct dependence most of the time is because the medium of communication (microphone/speakers and/or just air) is virtually transparent to the user. However, the environment can affect the quality of presentation of the seminar/lecture, such as when there is poor lighting, defective audio, or a bad OHP. One of the design guidelines for VLH, thus, is to make the communication tools 'transparent' so that an 'interesting' instructor and a well-prepared presentation can still captivate the audience irrespective of the new communication media being used.

- The VLH software should try to create interfaces that would enable the user to transcend the absence of actual physical presence [11] and the consequent inability to follow norms that are natural ([5],[1]) to a real Lecture Hall.

- The interfaces should have flexibility, parameterizability (offering a range of alternative behaviors that users can select) and tailorability (allowing users to make changes to the system itself) [10], [32].

- The system's response must be situated in the same sense, as is the user's activity [10]. The system should convey a homogeneous metaphor through its various interfaces and responses [32], [22].

- The system should lend itself to customizations of both function and presentation [10].

- Given the ubiquity of communication tools available on the World-Wide Web (WWW), 'reinvention the wheel' is to be avoided [26]. The software to be built is to make use of the available tools to the maximum extent possible.

4.3 VLH Design Details

The design of the VLH is presented in terms of the networking framework underlying the software, the interfaces planned for the software and the interactions that we perceive are possible using this framework and these interfaces.

4.3.1 Underlying Framework

The framework provided for the VLH software has a paradigm as shown in Figure 4.1.

Each learner's *VLH* software makes four network connections (one for each of the 4 client components it is made up of) to the respective servers that are part of the instructor's software. The model of communication supported by this framework is one-to-many (broadcast) from instructor to learners and one-to-one the other way. One potential drawback of the framework is that, if the right kinds of tools are not available, the server will be swamped by network connections when a large number of learners connect to the servers. This could be avoided by using tools that are based on the Multicast paradigm. However, we could not find multicast tool packages suitable for the purpose at hand. The issue of swamping of the server by network connections and other networking/performance considerations are not dealt with in this work.

4.3.2 Interfaces and Interactions

Two interfaces (referred to as "*VEIL* interfaces") have been designed to be used one each on the server- (instructor-) and client- (learner-) sides. The *VLH* interfaces include the

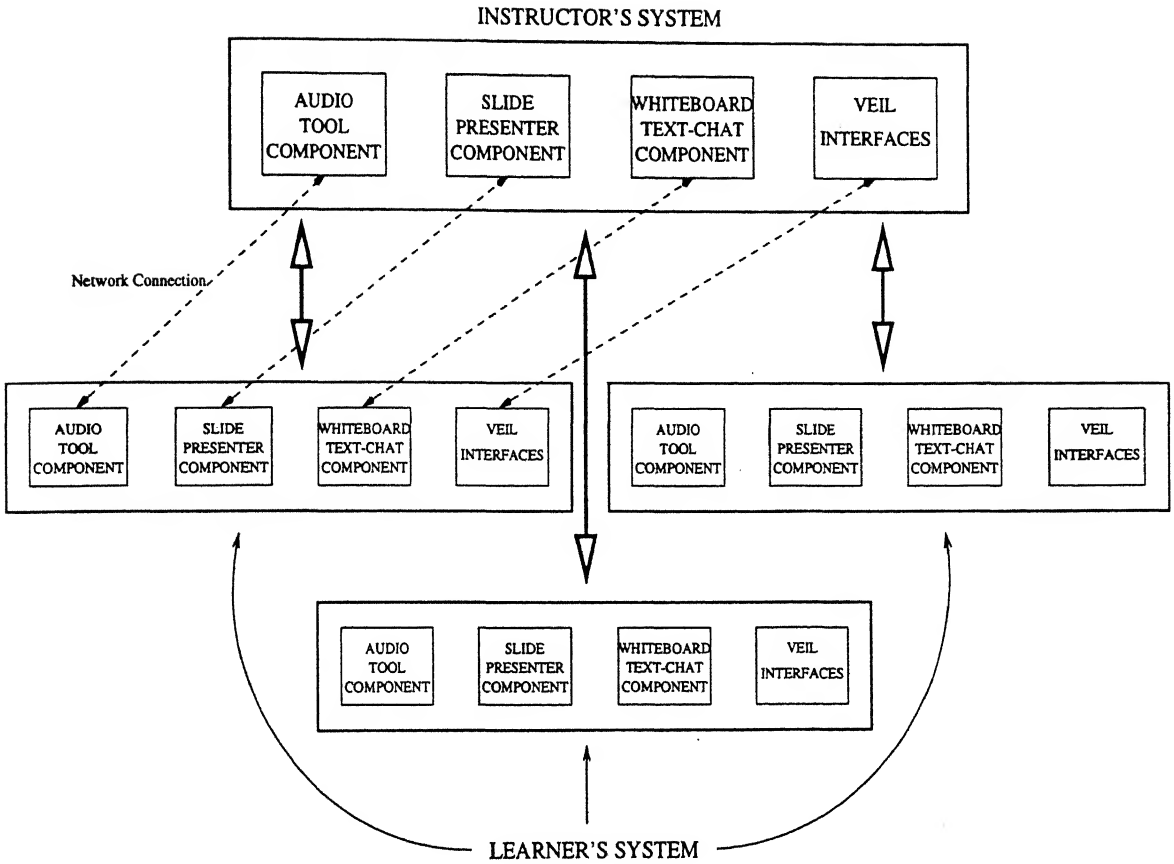


Figure 4.1: FRAMEWORK FOR THE VLH SOFTWARE

interfaces of the communication tools, that come in-built with the other tools to be used. In addition to these, the *VEIL* interfaces provide control and status information. These interfaces enable the users to follow the norms of the environment and give them a greater feeling of control over the environment. Figure 4.2 shows a possible configuration of the screen on the instructor's side. The various windows on the screen and the interactions they facilitate are:

- Slide Presenter:

  This window acts as the OHP equivalent in the Virtual Lecture Hall. It has controls to show the next and the previous slides, to point to a particular item on the slides, to zoom a portion of the slide and to type some text. These are the major operations that an instructor needs to perform while making a presentation. An additional capability

that can be added is enabling the instructor to draw also. This would give it full whiteboard functionality. A whiteboard can be provided in addition to the slide-presenter with screen capture facility. This would help in the process of elucidating a particular point of the presentation.

- Audio Control:

  This provides the user to control the audio tool. Apart from ON/OFF facility, it also provides for a listen-only facility. The listen-only facility is very much essential as pointed out by [1]. The audio of the audience is kept in the OFF state while the seminar is in progress. When they desire to interrupt the proceedings or when there is a question-answer session going on, they make a signal (a special control sequence) through their Text Chat window. The instructor then decides who (s)he will turn his/her attention to.

- Text Chat:

  On this window, a user-specified number of the latest statements by the various persons participating in the audience appear. This tool is useful during the initial stages of setting up the package and in establishing protocols and whenever there a clarification is requied about the other tools or about the norms of the environment.

- Control Panel:

  This panel gives the instructor information about the audience and the status of their interactions at-a-glance. It has three columns with the first column for the name of the user entries for name. The second column shows, with the help of an appropriate icon (a colored light), if a message from the user is currently on the Text Chat window. As the Text Chat window gets updated, this information is also updated. The entries in this window are sorted in a latest-first order and displayed accordingly. The third column in the Panel shows if the user has requested to interact via audio. An appropriate icon (a 'red light' glowing) is shown is shown on the screen. To enable a particular student to speak the instructor clicks on the icon (colored light). The Control Panel conveys a special code to the learner-side, where a similar interface conveys the information that specifies which student should speak. That information is displayed to the student on the Student Interaction interface(s).

  A facility needs also to be provided so that the instructor may specify that the audience should not interrupt the session to ask questions ('all questions at the end'). Another facility that can be incorporated into the control panel is for the instructor to inform

to the learners of his/her temporary absence from the seat or inability to give response (as when answering phone call).

- Visual Cue Window:

  This window is to make up for the lack of physical presence. When a normal lecture is going on it can have a photograph of a group of students. When at least one person wants to ask a question, as conveyed to the Text Chat server from the client side, the image of a raised hand with the words: 'Excuse me', can be shown. When the instructor clicks on a specific student in the Control Panel, the information can be conveyed to the Student Interaction panel. Then a small close-up image of the student can be shown on this window. This helps the instructor to ignore the absence of the person as (s)he directs his/her attention to the virtual presence on screen as the voice of the student is heard over the speakers.

Figure 4.2: INSTRUCTOR'S SCREEN

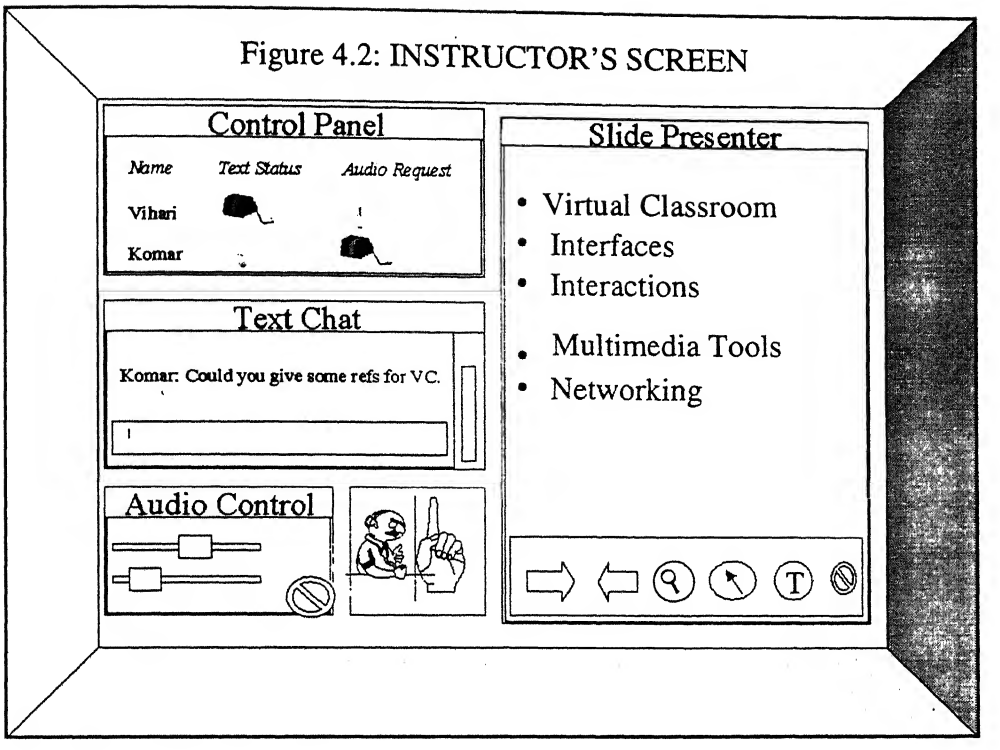
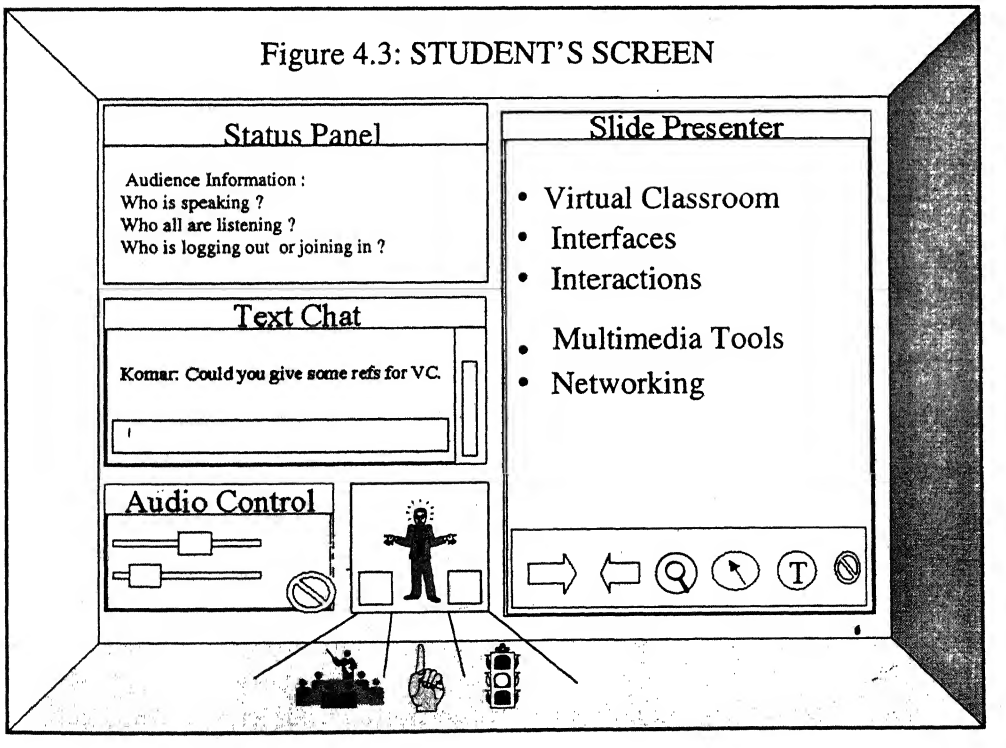


Figure 4.3: STUDENT'S SCREEN



A possible configuration of the Student's screen is shown in Figure 4.3.

The various windows on the screen and the interactions they enable are:

- Slide Presenter:

  This window functions similar to the corresponding instructor's window. Normally the student should not need to use the controls that may be provided as a part of the tool. If possible the instructor could be given the option of suppressing the controls of the students. If it is not possible, there should be a norm about the use of the slide presenter controls. The learner may be provided with access to the whiteboard so that (s)he may use it to explain a point or to ask a question.

- Audio Control:

  Apart from the usual controls to adjust the volume of the local audio tool, this could also provide controls to turn the tool off/on or use it in a listen-only mode. But taking into consideration the observations of Ackerman, [1] about the security and privacy issues while using audio, it would be ideal if the student's audio runs in listen-only mode all the time enabling him/her to speak only when the instructor is ready to listen to him/her.

- Text Chat Window:

  The function of this window is similar to that mentioned for the instructor-side text chat tool. The user types any special comments/questions into this window. The window is split into two parts at least, to demarcate the regions where the owner of the system types and where the output from other users who are chatting is shown.

- Control Panel:

  This provides the user the ability to interrupt the proceedings and to announce his/her availability to the other participants in the session.

  When the user types a special character sequence such as the Audio Request ('I want to speak'), a message is sent to the server. Also the information is passed on to the other users, so that it can be displayed on the Status Panel. That information is also used to display an 'Excuse me!' icon/message on the instructor's Visual Cue Window. A facility for the learner to cancel his/her Audio Request ('My doubt has been clarified, I don't have anything to ask.') needs also to be provided.

- Status Panel:

  This panel shows information about who all are part of the audience and who all

have requested for Audio. Such information helps the people feel in control of the environment [1]. When a user logs-off that information is also conveyed to the Student.

- Visual Cue Window:
  This window aims to help the user ignore the physical distance from the instructor in the Virtual Lecture Hall. Very often, virtual environments cause feelings of isolation resulting in loss of interest in the proceedings [11]. To avoid this, when a seminar is going on and the instructor is speaking, a long shot of the instructor with OHP and screen in the background can be shown. The choice of photograph can be left to the student. When the student wants to ask a question, and has pressed the special character sequence, an icon of the raised hand with the words 'Excuse Me' can be shown in this window. This confirms and reminds the student that his/her Audio Request is very much on and that he might get a chance to interact with the instructor soon. There is a problem with this one-to-one communication with the instructor the others do not get to listen to what is going on. It is up to the instructor to repeat the question addressed to him for the benefit of others or to device other means to get around this problem. The Text Chat can also be used when there is a pause in the audio that the Student receives, as the instructor listens to a particular student.

Tools that support all-to-all broadcasts are not yet conspicuously available on the WWW to be used for the purposes of VLH. When the instructor clicks in his Control Panel to enable a student to speak, appropriate messages can be shown on the Text Chat window.

Also, a close-up image of instructor can be shown on the Student's Visual Cue Window. This again helps the student to focus and feel that (s)he directly talking to the instructor. The process of requesting audio and speaking only when the student is given a turn helps reduce overlaps in conversation which are a great source of problem in an audio-only medium [1].

As Dourish [10] says, it is difficult to draw a line that signifies the end of the design process. While some of the main ideas of the design proposed here might remain, the interfaces are going to be redesigned in all probability during the process of implementation and use. Until the design achieves a certain amount of stability we decided to refer to the VLH software as a 'prototype'.

The implementation of *VLH* uses these design ideas in developing the interfaces. Some of the functionalities mentioned above under the names of some of the interfaces have been clubbed to form a new interface with a new name. This has been guided by ease of

implementation in the development environment. Further details of *VLH* implementation are presented in the next chapter.

Chapter 5

Implementation

In this chapter, the details about the implementation of the *VLH* prototype are given. This essentially involves the development of the *VEIL* interfaces, since software for the other tools, to be used as per the *VEIL* methodology, had already been gathered. The implementation details presented here include some background information about the development environment and about the tools used for the development of the *VEIL* interfaces as Visual C++ applications. A brief description about the objects that the software system comprises of, and a functional diagram explaining the interactions between the objects is also given. The *VEIL* Protocol that was designed and implemented to enable the instructor and learner to exchange messages is mentioned. An effort at testing the software prototype is also briefly touched upon at the end. Additional details are presented in Appendix B.

5.1 Development Environment

The VEIL interface software was built using Microsoft Visual C++ (Version 5.0) that uses Microsoft Foundation Classes (MFC), Version 4.0. The development work was done in the Microsoft Visual Studio integrated environment. The operating system used was Microsoft Windows 95.

Two PCs with Pentium processors, equipped with multimedia kits, connected via a LAN, were used for the development and testing. Some tests were also made across the campus-wide LAN at the IIT-Kanpur campus to test the quality of audio tool and the slide presenter in the presence of other network traffic. The results of the tests were encouraging. The multimedia kits included a SoundBlaster Audio card (16-bit, Full-Duplex), two 3-Watt

ampli-speakers, and a microphone each.

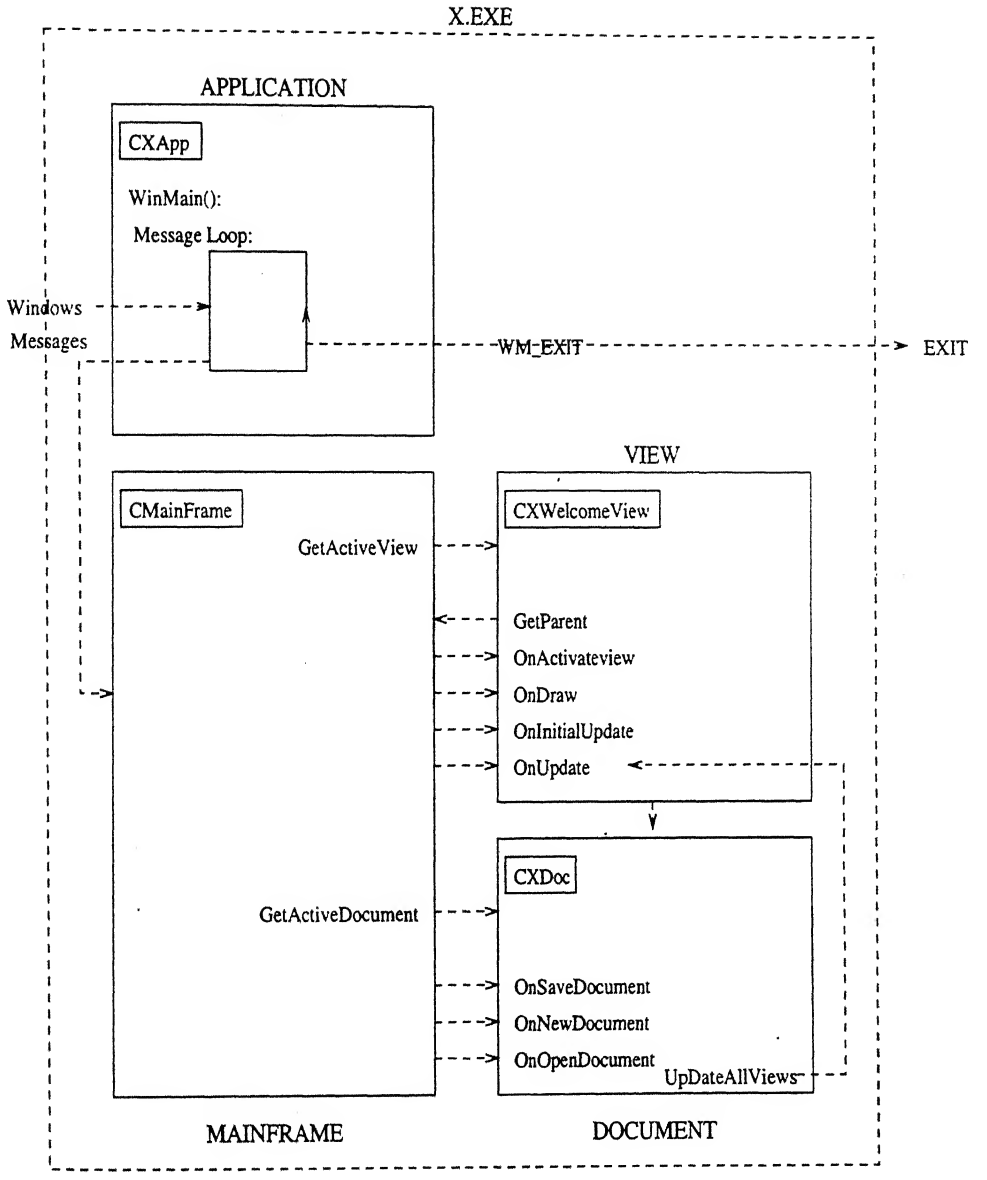
Microsoft Visual Studio is a package of software development tools that enables Rapid Application Development (RAD). It includes a support for the C++ language and provides tools for editing source files, compiling, linking, editing resource files like bitmaps, icons, cursors. It also provides an "application wizard" which takes specifications about the application to be built from the user through a series of dialogs and generates a skeleton application. This application as it is provides the user with a resize-able window that has some built-in menus (like the File Menu), a built-in tool bar and other regular window features. The user of Visual Studio, (the application developer) then modifies the existing menus, toolbars, adds interfaces/dialogs, and builds the actual functionality into the application.

Among the various pieces of software that can be created using Visual Studio is the MFC 32-bit (EXE) Application. This was the category of the application that VEIL interface prototype falls under. A brief description of the object-oriented structure of the MFC 32-bit (EXE) Application is given in the following section. An understanding of this will be useful in understanding the implementation details of the VEIL interface prototype (referred in the rest of the chapter simply as VEIL).

5.2 MFC 32-bit (EXE) Application Structure

An MFC (EXE) application has 4 main objects derived from standard MFC classes such as CWnd, CView, CApp. For an application "X" that we create, these four main objects automatically created are called: CXApp (the whole application object), CXMainFrame (the "frame" of the window), CXView (the view area of the Window) and CXDoc (the document object).

The relation between the various objects is shown in the figure below [15].



Typically, the contents of the Document are displayed in the View of the application. In other words, the View can be thought of as a window into the contents of the Document. In an SDI (Single Document Interface) application (which VEIL is), there is only one View peeping into the Document contents. But in a MDI (Multiple-Document Interface), there may be many Views showing different parts of the Document contents.

5.3 VEIL Application Structure

In addition to the 4 objects that the Visual Studio Application Wizard created, VEIL uses 7 more objects for the instructor-side application (called Veil:Instructor) and 6 objects for the learner-side application (called Veil:Student).

The additional objects that are we include in each of the instructor-side and learner-side applications can be grouped into 3 logical headings: (i) GUI (Graphical User Interface), (ii) Networking, and (iii) Session Information. The objects are presented below according to these logical groups. The functionality of the application that is encapsulated in the VEIL objects is also mentioned.

5.3.1 VEIL Object Description

- GUI Objects

  - Begin Session Dialog (Instructor-side only):
    This is implemented as a modal dialog box (the user cannot use the rest of the desktop until the dialog is completed). It takes the name of the session and the instructor as inputs. When the instructor begins a session, preparations are made to listen for and accept connections from learners.

  - End Session Dialog (Instructor-side only):
    This marks the end of the session. It takes the input of a parting message from the instructor. Later, this message is sent to all the learners logged on and the network connections with the learners are terminated.

  - Login Session Dialog (Learner-side only):
    The learner types-in a login-id and password which are sent with the help of the Document objects member functions to the instructor for verification. On successful verification, a joining-in message is sent to the instructor.

  - Logout Session Dialog (Learner-side only):
    The learner leaves the session and leaves a parting message. This message is sent to the instructor before terminating the network connection to the instructor.

  - Status Panel Dialog
    This panel provides at-a-glance information about the session in progress: the name

of the session, total number of persons logged on, the names of the persons who are logged-on as learners, personal information about a particular person (where provided), the names of the persons whose request to interrupt the session (to ask a question or to clarify a point) are pending, the availability of the person is his/her seat.

On the instructor-side, the panel contains additional controls that enable the instructor to send a message to a particular person whose interrupt request is to be processed. Additional buttons also provide means through which the instructor can inform a person that his/her mike is on/off (when it should be otherwise).

- Running Commentary Dialog
  This panel provides a running commentary of the whole session. The running commentary is provided as verbal descriptions of the various interactions (called events) that occur during the session. It lists the two latest events in the panel and provides means by which a history of events of the session can be viewed serially or searched through.

In addition to the above dialog boxes, appropriate icons were designed using the editors provided for the tool bar that is displayed as a part of the Mainframe. The details of these interface objects developed are shown in Appendix B as images captured from the screen. A set of accelerator (short-cut keys), drop-down menus and click-able icons provide easy ways in which the various interactions can be signaled.

Networking Objects

- Listening Socket (Instructor-side only):
  This object comes into existence when the instructor first begins the session. This socket goes into a "listen-mode" and waits for learners to send a "connect" message. For each connect message it receives, it causes another socket, the Client Socket, to be created. The Client Socket is paired to the socket of the learner to establish a network connection.

- Client Socket (Instructor-side only):
  This object is created whenever a learner requests a connection. It receives messages from the learner and causes them to be processed. After a message is processed, if

required, it is sent over the other client sockets to all the learners. It also keeps track of the messages that have been sent on the socket.

- Veil Socket (Learner-side only):
  This object is created when the learner wants to join a session. It contacts the Listening Socket object of the instructor machine using a pre-decided port value and the IP address of the instructor machine. Once a connection is formed, all further sending and receiving of messages occur through this socket object.

- Message
  On the instructor side, this object is used as a repository of all the messages received from the learners. On the learner side, this object is used to hold the outgoing message as it is sent or the incoming message while it is processed.

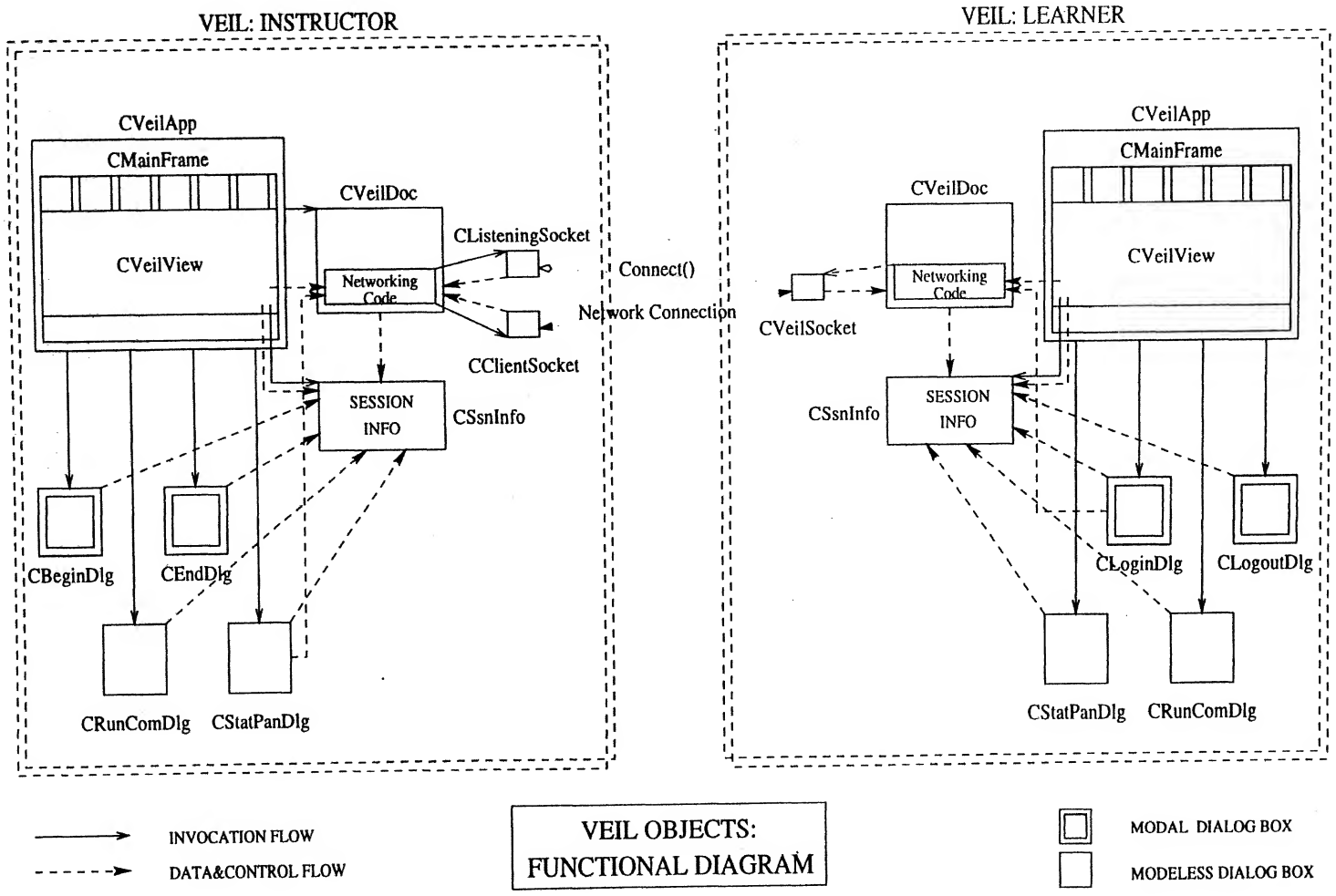
- Session Information Object

  - SsnInfo
    This object on both the instructor- and learner-sides holds all the information about the session in progress: the arrays of names of persons logged in, various flags that indicate various statuses, session name, instructor name and the like. This information is used by the GUI objects, the networking objects, and also by the CVeilDoc and CVeilView objects. It is a global object, a pointer to which is embedded in each of the objects that use it.

The CVeilDoc and CVeilView objects which are automatically provided by Visual Studio are modified substantially to incorporate the networking functionality of the application.

5.3.2 VEIL Object Functional Diagram

The functional diagram of the objects of the application is given below. It shows how the seven(six, for the application on the learner-side) VEIL objects are invoked from the 4 main automatically-generated and suitably-modified objects of the application. It also shows the various interactions between the objects that result in data and control flows that achieve the implementation of VEIL Design.


VEIL: INSTRUCTOR
CVeilApp
CMainFrame
CVeilView
CVeilDoc
CListeningSocket
Networking Code
CClientSocket
SESSION INFO
CSsnInfo
CBeginDlg
CEndDlg
CRunComDlg
CStatPanDlg
Connect()
Network Connection
VEIL: LEARNER
CVeilApp
CMainFrame
CVeilDoc
CVeilView
Networking Code
CVeilSocket
CSsnInfo
SESSION INFO
CLoginDlg
CLogoutDlg
CStatPanDlg
CRunComDlg
INVOCATION FLOW
DATA&CONTROL FLOW
VEIL OBJECTS:
FUNCTIONAL DIAGRAM
MODAL DIALOG BOX
MODELESS DIALOG BOX

5.4 VEIL Protocol

The various commands that the learners can invoke are linked to routines that send specific message packets across the network to the instructor. The instructor processes each incoming packet, modifies it if necessary, and resends it to all the learners as appropriate. The instructor may also send some message packets by him(/her)self which are sent to all the learners. The machines that run the VEIL applications process these packets, appropriately displaying messages on-screen, modifying the interfaces objects, or updating the Session Information data structures.

All these message packets exchanged between the VEIL applications follow a protocol, called the VEIL Protocol. The messages of the VEIL Protocol are:

1. *LOGIN:* The learner sends this along with login and password strings. Instructor verifies it and modifies to inform the learner that login was successful before forwarding the packet to all the learners logged on. Other learners update their session information data structure and add this learner to their list of logged users.

2. *LOGOUT:* The action that results is similar to the one for LOGIN, except that this message causes a deletion from lists of other users and a closing of the network connection on the learner and instructor side.

3. *REQINTR:* The learner sends this message to the instructor when (s)he wants to draw the attention of the persons participating in the session to a particular tool informing them that (s)he wants to interrupt the session to ask a question or clarify/make a point. This message is forwarded to all the learners after the instructor's application updates its session information appropriately. All the learners' applications also update their session information likewise. This information is shown in the Status Panel.

4. *CANCELINTR:* This message is sent by the learner to cancel his/her pending interrupt. This message causes appropriate updation of session information data structures and interfaces at the instructor- and learner-sites.

5. *ALLOWINTR:* This message is sent by the instructor to all the learners to inform the first learner (default choice), or another chosen learner, whose interrupt is pending that it is his/her turn to speak.

6. *DENYINTR:* This message is triggered by the instructor when (s)he wants to inform/request the learners not to interrupt the proceedings. All pending interrupt requests will be cancelled.

7. RESUME: This message is sent by the instructor for the first time when the session actually begins. (When the instructor uses the Begin Dialog, it only causes the Listening Socket to be created. For a period of time, the instructor may like to wait for the learners to join the session before actually beginning information delivery.) The *RESUME* message is also sent by the instructor soon after a learner finishes interrupting the session.

8. *OUTOFSEAT:* This message is used by the instructor and the learner to inform all the participants in the session about the his/her availability in their seat.

9. *MIKEON:* This message is used by the instructor to inform a learner in case his/her microphone is on or off when it should be otherwise.

10. *EVENT:* For each of the above messages, a verbal description giving further details is generated and sent to all the users. These messages are called "events". The Running Commentary Panel displays these events.

11. *END:* This message is sent by the instructor to close the current session. This causes all the network connections to be closed.

5.5 Testing

The *VEIL* concept and its implementation in the *VLH* software prototype was tested by using the components of the system in two configurations. First, the *VEIL* interface component was not invoked and the other tools were used by themselves. A session was run in this manner with 4 persons participating in it (three from the Computer Science Lab using three different machines and one from the Telematics Lab across the campus). The issues mentioned in [5], [1] about the use of audio alone for computer-mediated communication were in evidence. There were a number of overlaps in conversation. Very often, the participants did not understand what was going on and had to resort to text chatting to clarify about the proceedings. Control over the tools could not be passed smoothly and once again the text chat tool was used to control the tools. The audio suffered from slight delays which added to the confusion.

In the second configuration, all the components were invoked. This caused a definite decrease in confusion about the use of the tools. The new norms that involved using the *VEIL* interfaces needed some getting used to. But there was a significant increase in the smoothness of the interactions.

It is to be noted that owing to the nature of the problem addressed, the tests for checking the effectiveness of the idea are essentially subjective. They depend upon the subjective judgment of the user of the system. This observation is elaborated upon in the evaluation section of the concluding chapter.

Chapter 6

Conclusion

In this chapter, the thesis work is summarized, followed by an attempt to evaluate the work in terms of its limitations and its significance. Some situations where the software built could be used are mentioned. The chapter ends by mentioning some areas of future work that can be undertaken.

6.1 Summary

In this report, we present the problem of synchronous information delivery over the Internet – a problem that has not been addressed by researchers adequately. We summarize issues that arise from the use of the Internet for distance education. These issues become very important specially when multimedia tools are used by multiple users located at physically distant locations. The majority of problems that affect the effectiveness of this mode of education are to do with human factors that arise in the process of computer-mediated communication. We also present some findings from the fields of computer supported co-operative work, human-computer Interaction and the areas of media Spaces, interfaces. We explain the concept of a *Virtual Environment for Interactive Learning* to resolve some of the issues that arise in synchronous information delivery.

A methodology is also proposed to aid in building distance learning software using the *VEIL* concept in a time-efficient and cost-effective manner. The steps taken in actually designing one such software called "Virtual Lecture Hall *VLH*", are described to illustrate the algorithm of the methodology. The details of the design and implementation of the *VLH* software prototype are presented and explained. Development of the pair of interfaces

– VEIL:Instructor and VEIL:Student – was the major part of the implementation work. These interfaces were designed and implemented following the iterative refinement method of interface development described by [10].

A summary of the salient features of the proposed *VEIL* concept and methodology is as follows:

- The concept takes the distance between instructor and learner into special consideration while attempting to solve the human factors. By this it seeks to solve the persistent problem of loss of motivation of the distant learner in the learning process.
- The methodology is not just a simple "bundling" of tools [20], [31], [23].
- It creates an opportunity for learners' emotional involvement through its provision for the design of interfaces for this specific purpose.
- It avoids "reinvention of the wheel" in software development by advocating component re-use.
- It reduces product development cycle time in this process.
- The methodology also provides for the rapid customization of the collected tools to suit differing information delivery constraints and repeated use for various courses.
- The methodology promotes modularity. This is especially desirable since newer tools that provide more desirable features can simply replace the existing tools. The *VEIL* interfaces can be modified to suit them without much effort, if necessary.

A preliminary and subjective test done using the *VLH* package was reported. It was mentioned that the test gave positive results. The limitations of the test have been pointed out in the report.

A package of 3 components: SpeakFreely, NetMeeting and *VEIL* interfaces, resulted out of this work. This package could be used, as a whole, for information delivery in a variety of situations for which a real-world Lecture Hall would be used. Some of these are:

- delivery of courses to adult learners
- paper presentations, seminars
- corporate board-meetings

- formal interview broadcasts with audience participation
- formal group discussions

6.2 Evaluation

A test was conducted to evaluate the usefulness of the interfaces. The result of the test was positive. However, it was observed in the previous chapter that the results of the tests were necessarily subjective. This makes an objective evaluation of the effectiveness of the *VEIL* concept a difficult task. A certain degree of objectivity can be achieved by using *VLH* over a period of time to actually deliver a course. During this period statistics about the interactions can be collected as was done in the study by Ackerman[1]. The number of overlaps in audio, the period of such overlaps, the number of times control passed from instructor to learners etc. could be logged and tabulated. This, in conjunction with the feedbacks of the users of the system during that period, could give a more objective evaluation of the prototype.

One of the major assumptions in development of the *VEIL* interfaces was the non-use of video. Video, it may be recalled was not used because the hardware and tools that provide this medium are as yet costly and not popularly used in the Indian context. Many of the problems with the norms would be solved through the use of video. That would make the *VEIL* interfaces developed as a part of the *VLH* software prototype unnecessary. The nature of interfaces that would be required when video is used would differ from the *VEIL* interfaces developed. However, for video to become common, the cost of video (hardware and network bandwidth) have to come down significantly. This may take a few years to happen. Till that time, the *VEIL* interfaces, that assume commonly available and affordable hardware, may offer a reasonable solution to deal with the human factors in Internet-based education.

6.3 Future Work

The field-study of an actual use of the *VLH* prototype to deliver a course, as suggested in the previous section, can be taken up as an extension to this thesis. Finding out objective measures of evaluation in such field-studies could in itself be another beneficial offshoot.

Extended use of the software prototype is also a necessity so as to suggest refinements

in the details of the interfaces built for the *VLH* software. Finding out these refinements and implementing them is an integral part of the future work that can be undertaken.

Another task that can be performed is the testing of the *VLH* tools over the Internet. The documentation that accompanies SpeakFreely, the audio tool, assures that the tools works with "telephone quality" over a 28.8 kbps modem and using a Pentium-based PC. This can be verified. SpeakFreely does not function across the firewalls present in the network. A tool that performs well on the Internet even in the presence of firewalls can be developed. Developing such a tool with inbuilt interfaces that use the *VEIL* concept (facilitating norms) can be a large project of its own.

Bibliography

[1] Ackerman, M., Hindus, D., Mainwaring, S., and Starr, B. Hanging on the wire: A field study of an audio-only medi a space. *ACM Transactions on Computer-Human Interaction 4*, 1 (1997), 39–66.

[2] Aktan, B., Bohus, C., Crowl, L., and Shor, M. Distance learning applied to control engineering labora tories. *IEEE Transactions on Education 39*, 3 (1996), 320–335.

[3] Allen, N. Designing an electronic writing classroom. *IEEE Transactions on Professional Communication 39*, 3 (1996), 232–238.

[4] Baldwin, D. Three years experience with gateway labs. In *Proceedings, Joint Conference of ACM SIGCSE/SIGCUE* (1996), ACM, pp. 6–7.

[5] Bly, S., Harrison, S., and Irvin, S. Media spaces: Bringing people together in a video, audio and computing environment. *Communications of the ACM 36*, 1 (1993), 30–46.

[6] Chiricozzi, E., Mancini, F., Paladin, G., and Ruggieri, M. Procedures and classroom architectures for the development of tele-teaching activities. *IEEE Transactions on Education 38*, 1 (1995), 83–89.

[7] Chou, C., and Sun, C. A computer-network-supported cooperative distance learning system for technical communication education. *IEEE Transactions on Professional Communication 39*, 4 (1996), 205–214.

[8] Davis, J., and Smith, T. Computer-aided distance learning, part 1: Audiographic teleconferencing, interactive satellite broadcast&technical japanese instruction from the university of wisconsin- madison. *IEEE Transactions on Education 37*, 2 (1994), 228–233.

[9] Dawson-Howe, K. Automatic submission and administration of programming assignments. *SIGCSE Bulletin 28*, 2 (1996), 40–42.

[10] Dourish, P. Developing a reflective model of collaborative systems. *ACM Transactions on Computer-Human Interaction 2*, 1 (1995), 40–63.

[11] Dumont, R. Teaching and learning in cyberspace. *IEEE Transactions on Professional Communication 39*, 4 (1996), 192–204.

[12] Dyer, D. Creating a virtual classroom for interactive education on the web. *http:/www.igd.fhg.de/www/www95/papers/62/ctc.virtual.class/ctc.virtual.class.html* (1996).

[13] Goubil-Gambrell, P. Designing effective internet assignments in introductory technical communication courses. *IEEE Transactions on Professional Communication 39*, 4 (1996), 224–231.

[14] Hartley, S., Gerhardt-Powals, H., Jones, D., McCormarck, C., Medley, M. D., Price, B., Reek, M., and Summers, M. Enhancing teaching using the internet. In *Proceedings, Joint Conference of ACM SIGCSE/SIGCUE* (1996), ACM, pp. 218–228.

[15] Holzner, S. *Microsoft Visual C++ 5: No Experience Required.* BPB Publications, 1997.

[16] Jorn, L., Duin, A., and Wahlstrom, B. Designing and managing virtual learning communities. *IEEE Transactions on Professional Communication 39*, 4 (1996), 183–191.

[17] Komaragiri, V. V., and Sanghi, D. Designing a virtual lecture hall software. In *Proceedings, SEARCC'98* (1998), Australian Computer Society, p. (accepted for publication).

[18] Komaragiri, V. V., and Sanghi, D. Interactive hypermedia courseware for the world-wide web. In *Proceedings, International Conference on Collaborative Networked Learning* (1998), Indira Gandhi National Open University, pp. 71–75.

[19] Marshall, A., and Hurley, S. Interactive hypermedia courseware for the world-wide web. In *Proceedings, Joint Conference of ACM SIGCSE/SIGCUE* (1996), ACM, pp. 1–5.

[20] Microsoft-NetMeeting. Internet conference tool (windows 95 version). *http://www.microsoft.com/ie/ie3/netmtg.htm* (1997).

[21] Miranda, J., and Pinto, J. Using internet technology for course support. In *Proceedings, Joint Conference of ACM SIGCSE/SIGCUE* (1996), ACM, pp. 96–100.

[22] Mohnkern, K. Visual interaction design: Beyond the interface metaphor. *SIGCHI 29*, 2 (1997), 11–15.

[23] Netscape-Communicator. Web-based package of conferencing tools. *http://www.netscape.com* (1998).

[24] Oakley, B. A virtual classroom approach to teaching circuit analysis. *IEEE Transactions on Education 39*, 3 (1996), 287–296.

[25] Penfield, P., and Larson, R. Education via advanced technologies. *IEEE Transactions on Education 39*, 3 (1996), 436–443.

[26] Pennell, R. Managing online learning. *http://www.nepean.uws.edu.au/cimit/staff/rp.html* (1996).

[27] PowerPlus. Ms powerpoint slide projection tool for the internet. *http://www.netscene.com/* (1997).

[28] Reilly, C., and Leplattenier, B. Redefining collaboration through the creation of world wide web sites. *IEEE Transactions on Professional Communication 39*, 4 (1996), 215–223.

[29] Sears, A., and Savet, K. Faq: How can i use the internet as a telephone? *http://www.northcoast.com/ savetz/voice-faq.html* (1996).

[30] SpeakFreely. Free audio tool with source code for windows. *http://www.fourmilab.ch/* (1997).

[31] Symposium. Web-based training delivery software. *http://www.centra.com/product/index.html* (1997).

[32] Tognazzini, B. *TOG on Interface.* Addison-Wesly Publishing Co. Inc., 1992.

Appendix A

Detailed Report of Literature Survey

Report of the Survey on Virtual Classrooms (VCs)

Aim of the survey:

Survey done to study ways in which a course (say, on Computer Networks) can be offered over the Internet, by integrating various software tools available over the World-Wide Web.

Survey covers approaches to deliver course material, to promote "classroom" communication, to give assignments & conduct quizzes, in other words, the activities of a Virtual Classroom. (Use of video for the Virtual Classroom has not been considered.)

Various alternatives, pitfalls, and tips from people's experiences in building and maintaining Virtual Classrooms (VCs) to be explored.

The findings of the survey -- the various issues encountered by educators, the solutions suggested/adopted by them, the various tools and techniques used -- are presented under the following headings:

* Course Delivery Mechanisms
* Course Material Preparation
* VC Communication
* VC Assignments and Quizzes

Additional information and references are given in appendices as follows:

* Appendix A.a: References (cited in this section)
* Appendix A.b: Tips, Insights, Advices
* Appendix A.c: Survey Details
* Appendix A.d: URLs and Additional References

A.1 Survey Report

Course Delivery Mechanisms

Infrastructure Decision. The first question to be answered is what is the infrastructure to be used for the VC? The answer to this question helps to determine the course delivery mechanism and the communication mechanisms that can be used.

The infrastructure decision in turn depends upon the following factors:

(1) the nature and content of the course
(2) the nature of the audience and the facilities available to them.

---

(1) The nature and content of the course.

The methodology that is to be used for teaching is an important consideration. For instance, courses such as those on technical writing involve more of self-paced learning and so a simple web of HTML documents and simple forms to turn in assignments can be sufficient. This in turn will require normal Internet connectivity, a browser and enough RAM on the user's computer to download long pages. In fact such a course can even be conducted by e-mail. [Some ideas about how a course conducted by e-mail can be enhanced are presented in Appendix A.c.] But suppose the course deals with *collaborative* Technical Writing [2], then annotation tools and probably a white board are a must.

The use of audio, video, graphics, animation are each necessary or appropriate with certain courses. A course can also be such that it can be delivered by using one or more of the above media. Where such alternatives are present, the richer medium (audio over text, video over audio)is to be preferred. The rationale behind is the concern among Internet educators about bringing a "personal touch" into the VC environment. However the use of richer media is constrained by the factor below.

(2) The nature of the audience and the facilities available to them

Since the course is offered across the Internet to students located at different places, the nature of their Internet connectivity, their local hardware and software resources (or at least the capability to download required utilities and use them) play an important role in deciding the medium of course delivery. Whether a class is held synchronously or asynchronously also affects the capability requirements.

For example: for a 1 hour video + audio lecture 40 MB hard disk space is recommended, if it is downloaded asynchronously and played later. And it may take 15 to 25 minutes to download. If the playback is at realtime (synchronously, as the data is being downloaded) then 20 MB of hard disk is still required and in addition a lot of RAM and of course, good quality connectivity for the duration of the lecture are also required [7].

For audio to be of good quality a modem of 14.4 kbps is a minimum. Even with this, compression techniques (such as "GSM") have to be used. Audio typically requires a 20MHz 386 PC or faster. [13]. Also if audio is used special purpose hardware (sound cards, speakers, microphones) and software (compression utilities, audio players or more often tools that bundle these) are required.

In most of the work reported in the journals referred, the emphasis is on course material presented as hypertext documents [2, 5, 7, 8, 10]. Very few report work that uses audio [3]. However there has been a surge in the number of audio tools being provided for "Internet Telephony". These tools are being used more and more in VC applications [13]. Results of a survey of some of these tools is presented later on.

Course Material Preparation

Even if other media are used for course delivery, it is essential to maintain a repository of course documents. A student who misses an on-line lecture, for example, would want to go through notes of the missed lecture.

Since a lot of work had been done with hypertext courseware, there is a wealth of advice on this aspect of courseware preparation. Methodologies and software which help to construct a web of hypertext & hypermedia documents have been developed. [11] presents a design methodology as a simple top-down algorithm that starts from problem definition and ends up with a well-connected hypertext web.

A whole lot of HTML "style-guides" and conversion tools (conversion from various word processor formats to HTML and other formats) are available at various locations on the WWW. (Appendix A.d contains URLs for some of these). Other tips and insights gathered are presented in Appendix A.b.

VC Communication

Need

This is another aspect of VCs that attracts a lot of attention from researchers. The students' motivation to pursue a course and hence the success of the course are linked with the extent to which the student feels "in touch" with the Instructor. Researchers studying the psychological aspects of Internet pedagogy say

that Internet students tend to feel "isolated". Even simple solutions such as including the instructor's photograph on the Web page and providing a profile, go a long way in reducing the dehumanizing effects of VC and bring them closer to normal classrooms [8]. (A mapping of activities in normal classrooms that can be simulated in VCs is given in [2].)

Apart from these psychological aspects, there are other concrete needs, such as clarifying the students' doubts, answering questions about the course etc. which are to be addressed by the communication facilities of the VC. Communication also comes in handy to make administrative announcements, to familiarize the students with the environment and with each other. (Some user surveys regarding all these aspects of communication can be found in [3].)

Alternatives

Email

There are various alternatives for providing VC communication. The most common and easily available one is e-mail. E-mail satisfies all the above mentioned requirements of a VC communication medium. One area where it falls short is clarification of doubts and answering questions. Many of these many be common to more than one student, causing unnecessary duplication of effort in answering/ clarifying them. A way out of this is for the Instructor to pass a FAQ periodically [5].

Listservs, Newsgroups

If the class is large, other alternatives such as listservs &

newsgroups can also be considered. These have the advantage that there is more interaction among students at the group level. Very often the Instructor's burden in answering questions is reduced because students tend to solve their problems among themselves. This is especially true in classes held over the Internet since the participants usually have a whole range of expertise in the field of the course [8].

Audio

Interactivity can also be built into audio-based virtual classrooms, provided the audio software provides for many-to-many multicast or at least 2-way one-to-many multicast rather than 1-way one-to-many delivery of audio data. A problem with many-to-many multicast is that the instructor would not have a control over who speaks when.

One useful tool that can be integrated with an audio-based course delivery is a WHITE BOARD. The Instructor could illustrate a point, draw a diagram as a concept is explained and students all over get to see it. Recent audio tools such as Speak Freely, Cyberphone are integrating a white board and other utilities such as text chatting, user-to-user file transfer, along with the basic audio mechanisms. There is a healthy competition among various audio tool writers and so each tool comes bundled with some extra features. For example with POWPOW, the users could surf the WWW together, apart from speaking to each other. A related concept to surfing the WWW together is implemented in ALBATROSS where the Instructor can take the students through a guided tour of the hypertext web so that the student get lost in the maze of HTML documents.

Annotation Tools

Yet another type of communication tools are those for annotation purposes. A student who listens to a lecture may like to make a note on the document being shown. An annotation tool such as CoNote, YARN WEB helps to do that. Annotations can be private and public. A public annotation can be seen by all who are sharing the document (and this makes it a white board !). There are a whole lot of people who have written annotation software, and like with audio tools, they keep adding extra features like text-chatting and other facilities, thereby blurring the boundaries between various tools.

All of the above mentioned classes of tools are available on Unix, MAC, Windows/DOS platforms. Some of them are present on all three. URLs of some of these are given in Appendix A.d.

VC Assignments and Quizzes

Most of the work described on assignments deals with proprietary software. This has two reasons: (1) a point stressed w.r.t. assignments in VCs is that feedback to students should be prompt. This helps to increase the level of participation of the students [5]. (2) the assignment system has to be fool-proof.

The student either runs a client program that contacts the remote assignment/course server or (s)/he telnets onto the server (into a restricted shell) [9] to work on the assignments.

Questions are typically of the "what if" kind where the problems are presented to each student with a change of parameters [1]. Innovative approaches, those that are not possible in a normal class, are also described. For example, a student can make more than one attempt to solve assignment problems (which are easily

generated by the software) until he gets a satisfactory score. This way he is coaxed to spend more time thinking about the problems. Another approach involves giving the solutions also along with the problems, but with a twist. Some of the solutions are wrong and the student is supposed to point out the wrong solutions. This way thoroughness is also checked. This has the advantage that the amount of submitted matter is less, thereby making the job of checking easy. But it has the disadvantage that it is not fool-proof as it is.

Another way to conduct the whole process of assignment submission is to give users passwords and ask them to ftp their work to a central server into specific folders. These folders can later be checked and graded either manually or with a "course-controller" [4] software that automatically does most part of the checking work.

Quizzes that include multiple-choice questions can be conducted using CGI-(Common Gateway Interface-) scripts. Software is available that will help generate these cgi-bin scripts that are appropriate to handle the data sent via a form [12].

On the whole most Instructors prefer to conduct normal exams and use normal grading schemes rather than depend upon the automatic systems.

Appendix A.a: References

Note: Abbreviations Used

ITOE: IEEE Transactions on Education

ITOPC: IEEE Transactions on Professional Communication

CONF: Proceedings, Joint Conference of ACM SIGCSE/SIGCUE, 1996

---

[01] Baldwin D.
"Three Years Experience with Gateway Labs",
CONF, pages: 6-7

[02] Chou C. & Sun C.
"A Computer-Network-Supported Cooperative Distance Learning System for Technical Communication Education"
ITOPC, 39(4): 205-214, Dec 1996

[03] Davis J.L. & Smith T.W.
"Computer-Aided Distance Learning, Part 1: Audiographic Teleconferencing, Interactive Satellite Broadcast&Technical Japanese Instruction from the University of Wisconsin-Madison",
ITOE, 37(2): 228-233, May 94

[04] Dawson-Howe K.M.
"Automatic Submission and Administration of Programming Assignments"
SIGCSE Bulletin, 28(2): 40-42, June 1996

[05] Dumont R.A.
"Teaching and Learning in Cyberspace"
ITOPC, 39(4): 192-204, Dec 1996

[06] Hartley S., et al
"Enhancing teaching using the Internet",
CONF, pages: 218-228

[07] Harris D.A. & DiPaolo A.
"Advancing Asynchronous Distance Education using High-Speed Networks",
ITOE, 39(3): 444-449, Aug 1996

[08] Jorn L.A., et al.
"Designing and Managing Virtual Learning Communities"
ITOPC, 39(4): 183-191, Dec 1996

[09] Oakley B.
"A Virtual Classroom Approach to Teaching Circuit Analysis"
ITOE, 39(3): 287-296, Aug 1996

[10] Marshall A.D. & Hurley S.
"Interactive hypermedia courseware for the World Wide Web"
CONF, pages: 1-5

[11] Mengel S. A. & Capt Adams W.J.
"The Need for a Hypertext Instructional Design Methodology"
ITOE, 39(3), 375-380, Aug 1996

[12] Miranda J.E. & Pinto J. S.
"Using Internet Technology for Course Support",
CONF, pages: 96-100

[13] Sears A. & Savet K.
http://www.northcoast.com/~savetz/voice-faq.html

A.2 Tips, Insights, Advices

The following information was collected from the descriptions by people about their work on VCs. This may come in handy while implementing the VC.

The information is presented under broad topics connected with VCs. References given in square brackets refer to the papers in Appendix A.

* Registration/Setting Up

  Even though people use the Web, they are not all aware of basics like e-mail/FTP [5]. It is advisable to insist on some basic requirements from the student at the time of Registration.

* Extra Features

  Having a help desk can prove to be a life-line for users initially. They will need substantial help to familiarize them with the VC environment. Maintaining a repository of useful downloadable software is also a good idea [5].

* Users' Technical Problems and Preferences

Students using providers (like AOL, VSNL) have monetary cost to bear for the services and they cannot perform costly course functions [5].

Users may use different e-mail addresses or may change the addresses. This will cause problems while maintaining listserv lists and mailing lists. Having to make manual changes to the list can be very troublesome.

Most users do not read manuals or help web pages thoroughly. They do it just to get a working knowledge and to correlate previous unrelated things. This was found to be the cause of much confusion and needed many messages to and fro for clarification [5].

Confusion of metaphors "upload" "download" "server". Metaphors should be learner-centric.

In a survey of students who took a VC course, more than half of them preferred Asynchronous learning to MOOs since this is more flexible in terms of time scheduling [5].

Scheduling chats across time-zones is problematic.

Faculty should include their pictures and profile to make the students feel less isolated. [5]

* Implementation Details

FIREWALLs offer a problem for real-time playback.

Audio quality is more important than video & text when the

playback is synchronous. Audio encoding is problematic since it is less standardized across platforms than video. [7]

Users can be made to register a name and password of their choice before giving them access to the course material. This is to prevent unauthorized access of course material. [3]

* Course Preparation

Arrange documents logically. Structure and design of a Web Course differs from classroom course. Beware of Information overload [5, 11].

It is useful to have a link to a Glossary on all the course material pages. This will be especially useful when teaching totally new areas.

An idea is to put all the links from a HTML document to others at the bottom. This way students will not miss links and are given a greater chance to follow the logical progression of the course. There is no loss of continuity and no fear of missing some links.

* Communication

In audio systems it would be helpful for the Instructor to have control over who speaks. The software can possibly be tweaked so that on seeing that a student wants to speak, the Instructor can enable the audio of a student participating in a synchronous class.

Using mailing lists may not be a good idea for interaction,

especially if users have to pay by quantity of data sent/ received or for time used. Using the Course Home page as a bulletin board can be an alternative. This also has the advantage that unnecessary matter can be filtered out [12].

A list of FAQs and HowTos about general course functioning and subject topics should be made available to the students..

==========

A.3 Survey Details

The details of the survey are presented under the following headings:

- Basic Resources for teaching over the Internet
+ infrastructure
+ publishing
+ support tools
+ information sources

- Computer-Mediated Communication
  (from the point-of-view of Internet-based Education)
+ Asynchronous
+ Synchronous

- Assignments \& Quizzes

****
NOTE: The references cited below are given in Appendix A.d
****

Basic Resources

Infrastructure

Consists of instructor-side and student-side resources apart from the network connectivity.

The instructor-side resources include a server, technical support to maintain the server, WWW hierarchy, training and help desks for users.

Student-side resource requirements will depend upon the media used in delivering the instruction. At the minimum they will need a computer with enough RAM to hold long HTML pages. The other requirements include sound card (lot more memory and a hard disk if audiography/audioconferencing is used), modem (14.4 kbps minimum), speakers, microphones.

Publishing

* Conversion Tools

  Tools to convert a variety of existing word processor formats to HTML [L2HTML]. The quality of these tools may not be sufficient. Manual conversion may still need to be done.

* Authoring Tools

  WYSIWYG HTML editors (as with Netscape v3.0) and programs to

generate forms.

* HTML and link validation tools

Tools that check HTML pages to conform to standards and also to traverse entire WWW hierarchies. [SIMPLE] [Mengel]

* Scripting Tools

Common Gateway Interface(CGI) scripting using scripting languages like Perl can be used to process forms used for interaction. [CGI] [CGIGEN] [Perl]

Support Tools

* On the server side :

Tools that help to setup and maintain a WWW site such as -

+ Graphic and icon libraries for designing pages [Icons].

+ Search Engines that provide the ability to search all pages on the site for a particular word [Srch].

* On both the instructor and student sides :

+ A WWW browser (such as lynx, netscape)

+ Tools such as uuencode (uudecode) that can be used along with e-mail [UUDeview]

+ Tools to perform FTP (check if it is a part of the user's network software because in many cases it is a basic require-

ment.)

Given basic capabilities like FTP the user can be asked to download other support tools from a pool of shareware on the server.

Information Sources

Sources of information of interest related to the course can be provided as links to sites. These sources include books [Stevens], magazines, WWW pages, newsgroups, conferences and mailing lists [InfSrs].

Computer-Mediated Communication (CMC)

CMC means using Internet-based electronic communication technology that allows people to interact with each other both synchronously and asynchronously. This includes older, traditional, asynchronous forms such as e-mail and newsgroups/bulletin boards as well as newer, synchronous forms such as text conferencing, telephony, videoconferencing, graphics conferencing (shared whiteboards), audiographics (audio+graphics+(optional)annotation facility).

CMC can be either the main course delivery mechanism or can be used to enhance interaction among students, instructor and administrative staff. These possibilities are presented below for each medium.

Asynchronous

* E-mail

Mostly used for interaction between Instructor and students. But where Internet access via browsers is absent, it can be used along with supports tools like uuencode and uudecode to transfer binaries [UUdeview]. The binaries can then be used with the help of local software utilities (say an audio player software) to create the classroom atmosphere. The courseware (for e.g. the audio file) can then be mailed in the same way. The local utilites themselves can be downloaded from a mail-server that automatically processes incoming mail requests and "FTP"s them via e-mail.

* Usenet newsgroups(with limited scope) & Electronic Bulletin Boards

Used for announcements, discussions.

* Mailing lists (listserv, majordomo)

Used for announcements and for clarifying doubts and to circulate FAQs.[MailList]

* Downloaded Audio files

Where quality of connectivity is poor,the audio files (and graphics slides) can be downloaded via ftp and played locally.

Synchronous

---

* Audio Delivery Software

A large number of tools are available for transmitting audio across the Internet. The protocols implemented are not standard and so the tools may not interface with each other as a rule. A 14.4 kbps modem is a minimum and provides audio quality that is less than that of a normal telephone but good enough to be understood. The audio tool may distribute by actually multicasting the data or by simulating multicast.

Quite a few of tools offer additional features also like text-chatting, voice-mail, user-to-user file transfer. Software is available for Windows, MAC and Unix platforms.

Speak Freely (free) and Cyberphone offer full compatibility between Windows and Unix users. Speak Freely uses some of the popular standards like GSM compression and supports the VAT/RTP [VAT] standards.

CoolTalk (Netscape/Insoft): Netscape is now packaging an Internet telephony client with version 3.0 of their browser. The client includes a whiteboard, text chatting, voice mail and other unique features.
This is still in its infancy but once it is developed it is predicted that it will be used by a lot of people.[AUFAQ]

POWWOW is another highly recommended tool for "best personal communications" on the Web. It has text-based chat, voice communications, and users can even view pictures of each other.

VAT by Lawrence Berekeley Laboratories is another audio tool but being one of the earliest products it does not have any of the above-mentioned extra features.

* Annotation Tools

These are used in conjunction with shared spaces such as white-boards. Some systems also provide facilities for groups of people to make notes on arbitrary HTML documents. Facilities to make the annotations public or private (only the user sees them) are also present.

Annotation can also be used in audiographics, the instructor might annotate the slides to illustrate or explain a point by marking on it at his/her end and the marking appearing at the student's end at real-time. [Annot]

* MUD/MOO (Text-based Virtual Reality)

This is one of the talked-about paradigm for virtual classrooms. The user telnets onto a MOO (Multi-User Dungeon (MUD) Object-Oriented) server. He becomes an instance of an object called the MOO programmer.
Since he is an object he can perform specific operations that the object inherits such as creating other objects, or interfacing with other MOO programmers. Each MOO has a "theme". This theme determines the operations that the MOO user can perform. One of the first MOOs was LambdaMOO, which has a "community living" theme. That means when a user telnets onto the MOO server he can perform operations such as creating a house object for himself using appropriate operations.

Another theme is that of the IPL (Internet Public Library) a user who telnets onto the IPL MOO can perform all the usual operations that he can do when inside a real library. He can do text-based chatting with other people logged onto to the server like him at that time. At IPL Internet educators and researchers are supposed to meet and discuss things and share certain

resources made available to them on the server.

A lot of work has to be put into existing MOOs such as LambdaMOO to make them useful for VCs and very few people have used them for applications such as the one for which this survey has been done.

==========

A.4 URLs and Additional References

[Annot] Futplex : free software available
ftp://gewis.win.tue.nl/pub/futplex.
The author is Koen Holtman (koen@win.tue.nl)
/* Useful for Internet documents. */

/* A host of other Annotation systems are also available at
the home page of CoNote - "a small group annotation
experiment" at Cornell University. */

[AUFAQ] Jim Davis - davis@dri.cornell.edu
"FAQ: How can I use the Internet as a telephone?"
Version 0.5 - July 25 1996

/* Contains minimum specifications for audio communication
over the net and brief descriptions of various software
titles available for Audio communication, free and otherwise.*/

[AUDIO] Links to Audio Software
http://www.yahoo.com/Computers_and_Internet/Software/Internet/

[L2HTML] Latex2HTML : (Conversion Software)
ftp://src.doc.ic.ac.uk/packages/WWW/tools/translators/latex2html/
(Downloaded)
/* Also downloaded ps2gif and a list of other translators. */

[Perl] http://www.yahoo.com/
Computers_and_Internet/Programming_Languages/Perl

[SIMPLE] M. Hagler & B. Marcy
SIMPLE (software) : Helps to pull already designed and
implemented instructional material into a whole.
//http://gs1.cs.ttu.edu/simple/index.html

[Srch] http://www.yahoo.com/
Computers_and_Internet/World_Wide_Web/Databases_and_Searching/

[Stevens] Whole of Stevens Network Programming book is available
Chapter by chapter at
http://www.cs.tamu.edu/course-info/cpsc463/toc.html

[UUDeviw] Frank Pilhofer
UUDeview (software) : Unix, Windows versions
/* Operates on multi-file multi-part e-mails that
contain binaries. But unlike with uudecode,
it concats them figuring out out-of-order messages
and decodes them. Menu-driven. */
http://www.uni-frankfurt.de/~fp/uudeview/

[VAT] ftp://ftp.ee.lbl.gov/conferencing/vat
ftp://ftp.ee.lbl.gov/conferencing/wb
/* Lawrence Berkeley LABs FTP site. Also has VIC - a video tool.*/
/* Downloaded */

Related Papers

Note : Abbreviations Used

ITOE : IEEE Transactions on Education

ITOPC: IEEE Transactions on Professional Communication

CONF : Proceedings, Joint Conference of ACM SIGCSE/SIGCUE, 1996

---

[1] Huang C.
"An Enhanced Computer Networks Course"
ITOE, 38(3) : 279-290, Aug 1995

[2] Marti W.F. et al.
"PACKET tracing : a new paradigm for teaching Computer Networks Course"
CONF, pages : 162-164

[3] Orsak G.C. & Etter D. M.
"Connecting the Engineer to the 21st Century through Virtual Teaming"
ITOE, 39(2) : 165-172, May 1996

****

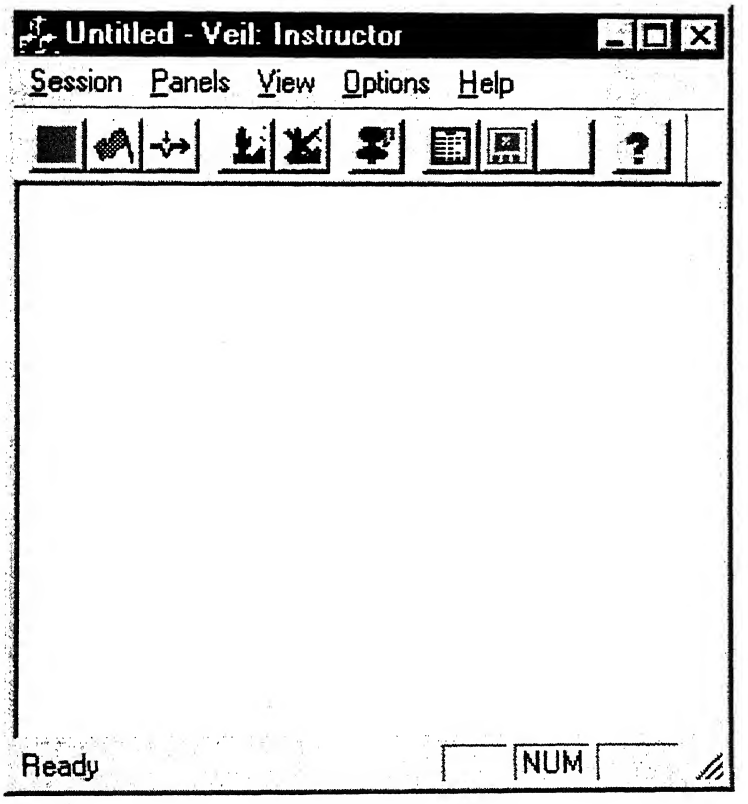
Appendix B

VEIL Interfaces Details

The details of the Interfaces designed and developed to implement the norms of the lecture hall are presented here as captured images from the screen. These interfaces and the functions that they are linked to, were developed with the aim of making Internet-based information delivery in an interactive (synchronous) learning environment more effective.

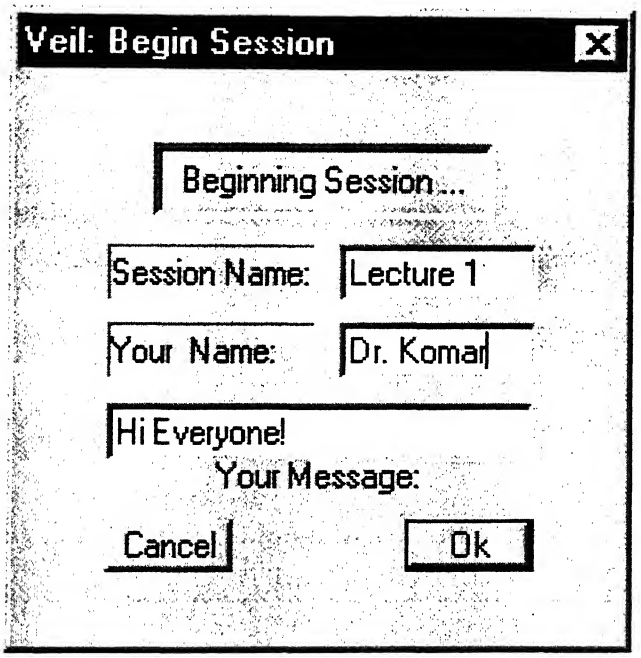
Presented here are:

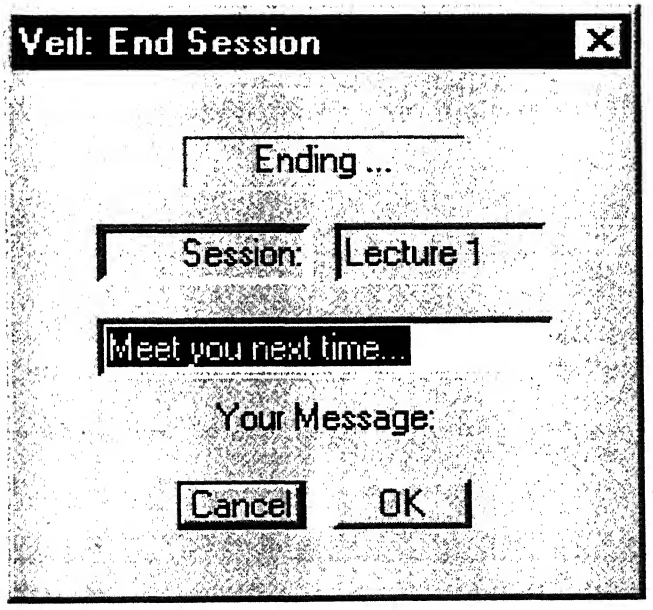
- *Veil:Instructor* – the Instructor's main application window
- *Begin Session* Dialog (Instructor's side)
- *End Session* Dialog (Instructor's side)
- *Status Panel* Dialog (Instructor's side)
- *Running Commentary Panel* Dialog (both sides)
- *Veil:Student* – the Learner's main application window
- *Status Panel* Dialog (Learner's side)
- *Login Session* Dialog (Learner's side)
- *Logout Session* Dialog (Learner's side)



Veil:Instructor - Instructor's main application window

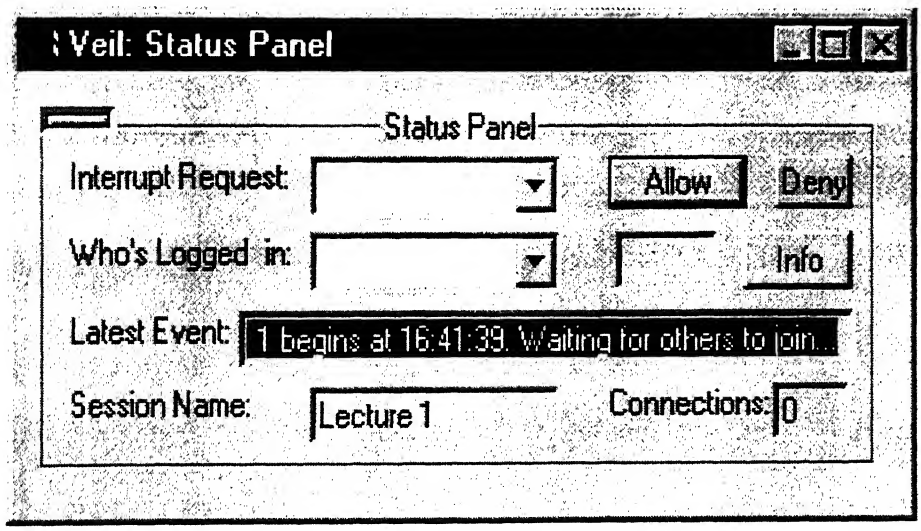
Begin Session Dialog (Instructor's side)

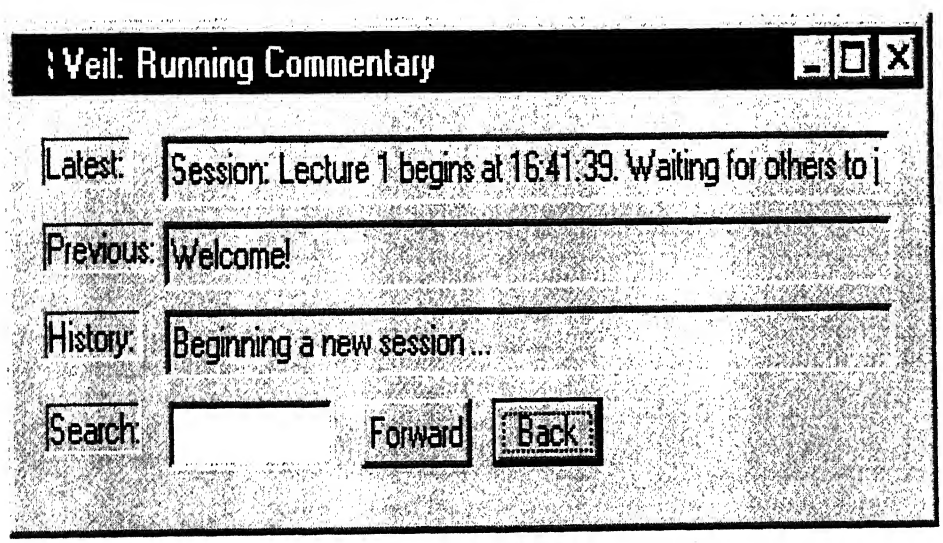




End Session Dialog (Instructor's side)

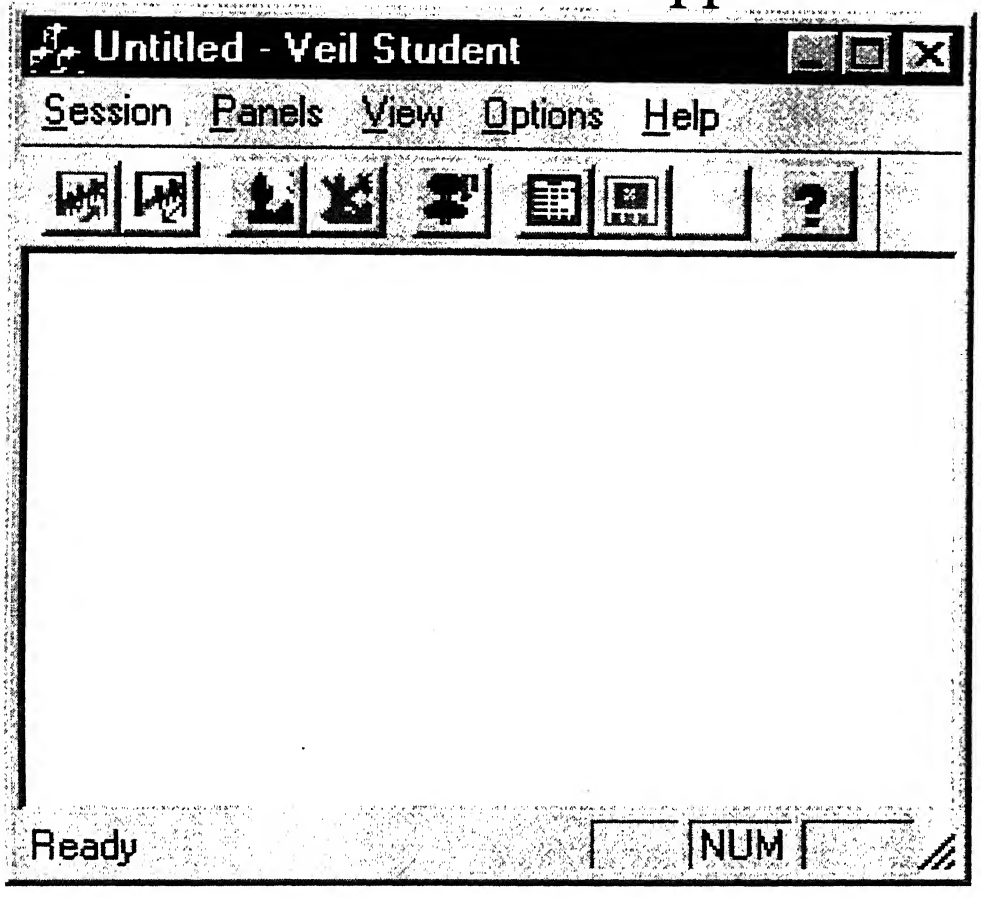
Status Panel Dialog (Instructor's side)

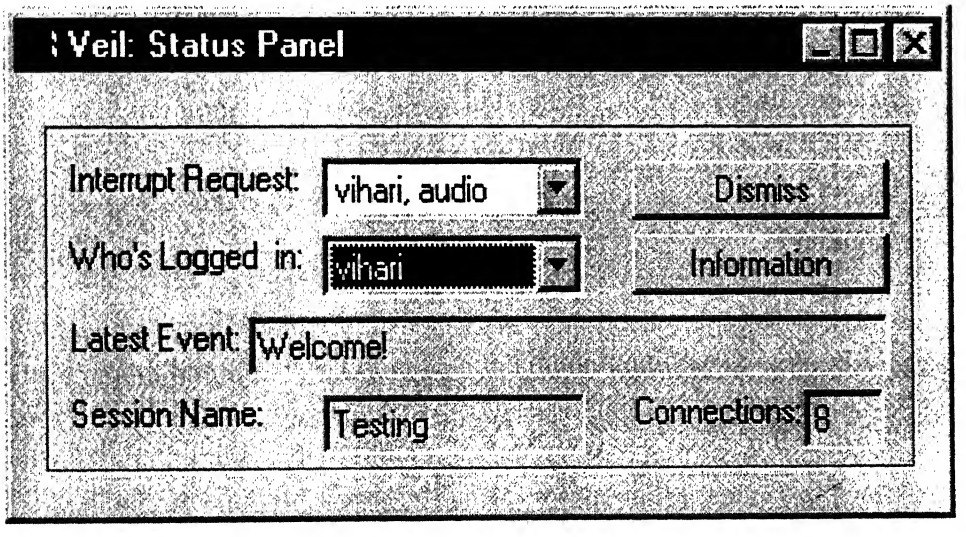




Running Commentary Panel Dialog (both sides)

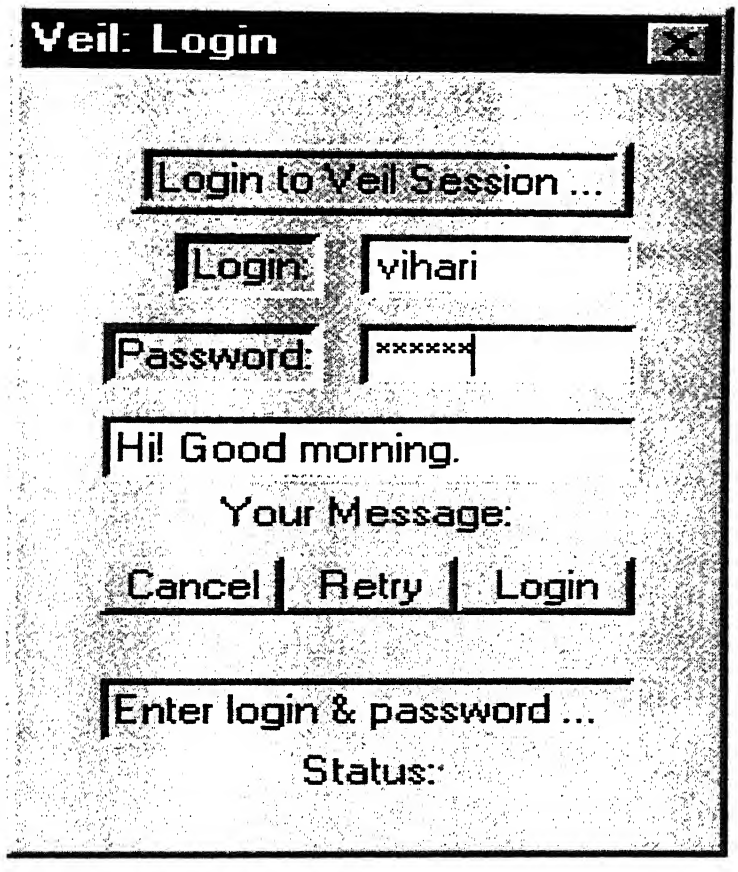
Veil:Student - Learner's main application window





Status Panel Dialog (Learner's side)

Login Session Dialog (Learner's side)



Logout Session Dialog (Learner's side)

Appendix C

Tips for Teaching in Synchronous Environments

The following tips about teaching from in a synchronous learning environment have been gathered from website of Symposium [31].

C.1 Tips for Teaching Online in Real Time

This document provides quick tips to make teaching with Symposium's real time environment more effective. The recommendations are things that we have learned from our pilot projects, and things that instructors have told us.

- **Use Interaction to Motivate, Engage, and Involve Learners**
  Facilitating Web-based training is like being the host of a very lively talk show. It is your job to keep your viewers motivated, engaged and involved. Web-based training programs delivered with Symposium are not passive experiences! To be successful, make learners part of the program by using the techniques outlined below.

- **Animate Your Delivery**
  The more excitement and energy that you convey through your voice, the more students will be motivated and energized. If you project your excitement, the participants will respond in kind. Be enthusiastic and a little louder than usual. This extra effort really does get them "up" for classes as well. This technique also works in the

classroom - when you show your interest, your students will be motivated to show theirs.

- **Engage Learners**

  Engage learners by asking them to participate verbally and intellectually. As a facilitator, the easiest way to engage students is to ask direct questions frequently. Ask learners to comment on a presentation, share their observations, or answer a direct question. Turn the tables by encouraging students to initiate questions to the instructor, as well as to other learners. Intellectually engage learners by asking them to think how the course is related to their experience and to consider other points-of-view.

- **Familiarize Yourself with the Course**

  If possible participate in the design of the course and become involved in the pilot. The more comfortable you are with the structure and content of the lessons in the course, the more time and energy you will be able to devote to facilitating the program.

- **Assess Learner Comprehension**

  Work with the course designer to include lots of short quizzes. These can be a great way to informally assess learners' comprehension. Quizzes can be true/false, multiple choice, or fill in the blanks Since true/false and multiple choice quizzes are automatically graded as the learners complete their quizzes, they get immediate statistical feedback. As the facilitator, you get a summary of these quiz results to guide the review of answers.

- **Ask for Informal Feedback**

  Use the Symposium "polling" feature to get feedback on things like the pace of the lesson and students' comprehension. In addition, the facilitator can poll the class by asking for simple yes/no responses. For example, the instructor can ask if students would like to continue the discussion.

- **Chunk Course into Short Segments**

  Chunk the content of your lesson into sections no longer than five to seven minutes. Use your objectives to identify discrete pieces of content. Once the content is chunked, use a variety of strategies to deliver it. For example, a program may have an outline as follows:

  – Minutes – Activity

- 0-2 Introduction and review agenda
- 3-6 Ask students to summarize key points from the last class
- 6-10 Present PowerPoint presentation "The Key to Service Excellence!"
- 10-14 Show video clip "customer scenario"
- 15-22 Run breakout groups to evaluate "customer scenario"
- 23-30 Debrief breakout groups using whiteboard
- 30-36 Interview Sales VP of North America
- 37-43 Field questions from audience regarding new sales plan
- 44-47 Remind students how to contact you and point them to threaded discussion and text chats
- 48-60 Sign off and encourage students to work asynchronously or synchronously for the last 12 minutes responding to topics in the threaded discussion or by participating in a text chat.

- Create Lessons with Multiple Media

  Use the power of Symposium to bring multiple forms of media into your lesson. Learning is enhanced when the message is delivered via text, images, sound. Select media elements that add value to your lesson such as a detail diagram showing how to install a new computer board. The diagram provides a visual explanation of the process while the facilitator describes the process. In this case the two channels provide complementary information that helps students better understand how to install a computer board.

- Vary the Interactivity

  Vary the interactions to keep learners attention. Lessons can include lectures, debates, role-plays, quizzes, question and answer sessions, Web Safaris and breakout groups.

- Draw on the Learner's Experience

  Asking learners to respond is a good way of breaking the ice, and making them feel more comfortable in the class. Learners often bring good insights of their own to the material and have a wealth of experience upon which to draw. Get everyone involved by asking questions like "Bill, we haven't heard from you this morning. How do you think this would apply to your situation?" When you call on someone by name like this, it personalizes things, too. And feeling personally connected to the class and the Instructor is important to good learning.

- Consider Using Humor

  While this may not match every instructor's style or subject matter, humor has been recognized as a great way to enhance learning retention. In live Web-based training, the tone of your voice convey the intended humor.